

Nom i cognoms:

Temps: 1h 30'

1. (2 punts) Els mètodes `pintaVaca()` i `pintaPatricio()` envien a pintar la geometria dels models d'una vaca i d'un Patricio. Les capses mínimes contenidores d'aquests models estan determinades pels punts: (`vacaxmin`, `vacaymin`, `vacazmin`, `vacaxmax`, `vacaymax`, `vacazmax`) i (`Patxmin`, `Patymin`, `Patzmin`, `Patxmax`, `Patymax`, `Patzmax`). El cap de la vaca mira cap a les X+ i el Patricio mira cap a les Z+.

Es vol visualitzar una escena formada per una vaca amb un Patricio a sobre. Ambdós models mirant cap a les Z+; el centre de la base de la capsa de la vaca ha d'estar situat al punt (10,0,0) i la seva alçada ha de ser 2; el centre de la base de la capsa del Patricio ha d'estar al punt (10,2,0) i la seva alçada ha de ser 1.

Suposant que la càmera està correctament inicialitzada (no cal que digueu res sobre la càmera), es demana que indiqueu, justificant les respostes:

- l'expressió de les TGs per ubicar cada model al lloc indicat de l'escena;
- el pseudocodi d'una funció `pintaEscena()` que pinti l'escena tal com s'ha descrit, calculant les TGs indicades i utilitzant `pintaVaca()` i `pintaPatricio()`.

Solució:

La vaca s'ha de rotar -90 graus respecte l'eix Y i s'ha d'escalar uniformement per a què la seva alçada sigui 2. El factor d'escala és doncs $2.0/(\text{vacaymax}-\text{vacaymin})$ (`escvaca = 2.0/(\text{vacaymax}-\text{vacaymin})`). Abans, però, s'ha de portar la vaca a l'origen, per tant portarem a l'origen el centre de la base de la capsa, que és el punt que després hem de traslladar finalment al punt (10, 0, 0). `cbvaca = ((vacaxmin+vacaxmax)/2.0, vacaymin, (vacazmin+vacazmax)/2.0)`.

El Patricio no s'ha de rotar, per tant apart de la translació inicial del centre de la base a l'origen (`cbpatr = ((Patxmin+Patxmax)/2.0, Patymin, (Patzmin+Patzmax)/2.0)`) i la translació final al punt (10, 2, 0), també caldrà escalar-lo per a què la seva alçada faci 1, és a dir factor d'escala $1.0/(\text{Patymax}-\text{Patymin})$ (`escpatr = 1.0/(\text{Patymax}-\text{Patymin})`).

Així doncs, les respostes a aquest exercici són:

- Expressió de les TGs de la vaca (TGV) i el Patricio (TGP):

$$TGV = T(10, 0, 0) * Ry(-90) * S(escvaca, escvaca, escvaca) * T(-cbvaca)$$

$$TGP = T(10, 2, 0) * S(escpatr, escpatr, escpatr) * T(-cbpatr)$$

- La rutina `pintaEscena()` en pseudocodi serà:

```
pintaEscena () {
    TGV = Translate (10,0,0);
    TGV = TGV * Rotate (-90, 0,1,0);
    TGV = TGV * Scale (escvaca,escvaca,escvaca);
    TGV = TGV * Translate (-cbvaca);
    ModelMatrix (TGV);
    pintaVaca();
    TGP = Translate (10,2,0);
    TGP = TGP * Scale (escpatr,escpatr,escpatr);
    TGP = TGP * Translate (-cbpatr);
    ModelMatrix (TGP);
    pintaPatricio();
}
```

2. (2 punts) Es vol veure l'escena de l'exercici anterior de manera que la vaca amb el Patricio mirin cap a l'observador, quedin centrats en el viewport, sense retallar i optimitzant l'espai que ocupen en el viewport. Indica TOTS els paràmetres d'una càmera perspectiva (posició+orientació i òptica) que ho faria possible. El viewport és de 600x400. Per facilitar els càlculs, en aquest exercici podeu considerar que les noves mides de les capses de la vaca i del Patricio un cop ubicats com toca són:

Patricio: dimensió en $X = 1$; dimensió en $Y = 1$; dimensió en $Z = 0.3$.

Vaca: dimensió en $X = 1$; dimensió en $Y = 2$; dimensió en $Z = 4$.

Solució:

El VRP el posem al centre de l'escena, que com que la vaca té la base centrada en el punt $(10, 0, 0)$ i el Patricio en el $(10, 2, 0)$, i la vaca té alçada 2 i el Patricio alçada 1, el centre dels dos objectes està al punt $(10, 1.5, 0)$, per tant **VRP** = **(10, 1.5, 0)**.

L'observador ha d'estar mirant cap a aquest VRP des de la direcció de la cara de la vaca, és a dir des del VRP en direcció paral·lela a l'eix de les $Z+$. Posem l'observador a distància 1.5 de la z_{max} de la vaca un cop transformada, és a dir **OBS** = **(10, 1.5, 3.5)**.

El vector up ha de ser la vertical de l'escena, així que **up** = **(0, 1, 0)**.

Per a l'angle d'obertura FOV, tindrem en compte on hem posat l'observador, com que hem posat l'observador a distància 1.5 de la z_{max} de la vaca un cop transformada (distància 1.5 d'on estarà el Z_{near} sense retallar) i l'alçada de l'escena és 3, tindrem que un angle que ajusta aquesta escena és **FOV** = **90 graus** ($M_{PI}/2.0$ radians).

El Z_{near} el posarem ajustat al principi de la vaca (z_{max} de la vaca un cop transformada) i per tant serà 1.5, i el Z_{far} per a què no retalli serà 5.5 (la vaca fa 4 de llarg). **$Z_{near} = 1.5$ i $Z_{far} = 5.5$** .

La relació d'aspecte ha de ser la mateixa que la del viewport per a què no deformi, així que **$ra = 600/400 = 3/2$** .

3. (1 punt) Per a visualitzar l'escena de l'exercici 1, un estudiant proposa la següent càmera: OBS=(20,0,0), VRP=(-9,0,0), up=(0,0,1) , ZN=15, ZF=40, FOV=90° i ra=1. Quan pinta l'escena no veu res en pantalla (malgrat que la rutina `pintaEscena()` és correcta). Quin paràmetre de la càmera creus que no és el correcte?

Considerem les mides de les capsas de la vaca i del Patricio, un cop ubicats, iguals que en l'exercici 2.

- a) vector up
- b) ZN
- c) ra
- d) VRP

Solució: b)

4. (1 punt) Cal definir una càmera a OpenGL; quin dels següents pseudocodis és correcte? Noteu que tant sols canvia l'ordre en què es fan les crides.

- | | |
|---|---|
| 1) VM=lookAT(OBS, VRP, up)
viewMatrix (VM)
PM=perspective (FOV, ra, zn,zf)
projectionMatrix(PM)
glViewport(...)
modelMatrix(TG)
pintaescena() | 3) VM=lookAT(OBS, VRP, up)
viewMatrix (VM)
PM=perspective (FOV, ra, zn,zf)
projectionMatrix(PM)
modelMatrix(TG)
glViewport(...)
pintaescena() |
| 2) modelMatrix(TG)
PM=perspective (FOV, ra, zn,zf)
projectionMatrix(PM)
VM=lookAT(OBS, VRP, up)
viewMatrix (VM)
glViewport(...)
pintaescena() | 4) glViewport(...)
VM=lookAT(OBS, VRP, up)
viewMatrix (VM)
PM=perspective (FOV, ra, zn,zf)
projectionMatrix(PM)
modelMatrix(TG)
pintaescena() |

- a) només 1) i 4) són correctes
- b) només 4) és correcta
- c) tots són correctes
- d) tots són correctes menys 2)

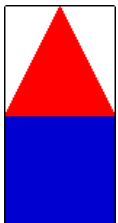
Solució: c)

5. (1 punt) Imagina que tenim l'escena de la vaca + Patricio de la pregunta 1 i els volem girar entorn l'eix Y (com si es tractés d'una peça d'uns cavallets –"tiovivo"–). Suposant que TG1 és la matriu de TG per ubicar la vaca i TG2 és la matriu de TG per ubicar el Patricio quin dels següents codis és correcte?

- | | |
|--|---|
| <p>a) <code>AUX= Rotate(alfa,0,1,0)</code>
 <code>TG1= AUX*TG1</code>
 <code>TG2= AUX*TG2</code>
 <code>modelMatrix(TG1)</code>
 <code>pintaVaca()</code>
 <code>modelMatrix(TG2)</code>
 <code>pintaPatricio()</code></p> | <p>c) <code>AUX= Rotate(alfa, 0,1,0)</code>
 <code>TG1=TG1*AUX</code>
 <code>modelMatrix(TG1)</code>
 <code>pintaVaca()</code>
 <code>TG2=TG1*TG2</code>
 <code>modelMatrix(TG2)</code>
 <code>pintaPatricio()</code></p> |
| <p>b) <code>modelMatrix(TG1)</code>
 <code>pintaVaca()</code>
 <code>Rotate (alfa,0,1,0)</code>
 <code>modelMatrix(TG2)</code>
 <code>pintaPatricio()</code>
 <code>Rotate (alfa,0,1,0)</code></p> | <p>d) <code>AUX= Rotate(alfa, 0,1,0)</code>
 <code>TG1=AUX*TG1</code>
 <code>modelMatrix(TG1)</code>
 <code>TG2=AUX*TG2</code>
 <code>modelMatrix(TG2)</code>
 <code>pintaVaca()</code>
 <code>pintaPatricio()</code></p> |

Solució: a)

6. (1 punt) Tenim una piràmide de base quadrada de costat 5, amb la base centrada al punt (0,0,2.5) i alçada de la piràmide 5 amb l'eix en direcció Z+. A l'escena tenim també un cub de costat 5 centrat a l'origen. El viewport està definit amb `glViewport (0,0,400,800)`. Si a la vista es veu la imatge que teniu al dibuix (caseta), quines inicialitzacions d'una càmera axon mètrica (posició+orientació i òptica) permetrien veure aquesta imatge? Tots els angles estan en graus.



- | | |
|---|--|
| <p>a) <code>PM=perspective (90, 1, 5, 10);</code>
 <code>projectionMatrix (PM)</code>
 <code>VM=translate (0,0,-10);</code>
 <code>VM=VM*rotate (90,1,0,0);</code>
 <code>VM=VM*translate (0,0,-2.5);</code>
 <code>viewMatrix (VM);</code>
 <code>pinta_escena ();</code></p> | <p>c) <code>PM=ortho (-2.5, 2.5, -5, 5, 5, 10);</code>
 <code>projectionMatrix (PM)</code>
 <code>VM=translate (0,0,-7.5);</code>
 <code>VM=VM*rotate (-90,0,0,1);</code>
 <code>VM=VM*rotate (90,0,1,0);</code>
 <code>VM=VM*translate (0,0,-2.5);</code>
 <code>viewMatrix (VM);</code>
 <code>pinta_escena ();</code></p> |
| <p>b) <code>PM=ortho (-2.5, 2.5, -5, 5, 5, 10);</code>
 <code>projectionMatrix (PM)</code>
 <code>VM=translate (0,0,-7.5);</code>
 <code>VM=VM*rotate (90,0,0,1);</code>
 <code>VM=VM*rotate (90,0,1,0);</code>
 <code>VM=VM*translate (0,0,-2.5);</code>
 <code>viewMatrix (VM);</code>
 <code>pinta_escena ();</code></p> | <p>d) <code>PM=ortho (-5, 5, -5, 5, 5, 10);</code>
 <code>projectionMatrix (PM)</code>
 <code>VM=translate (0,0,-7.5);</code>
 <code>VM=VM*rotate (90,0,0,1);</code>
 <code>VM=VM*rotate (90,0,1,0);</code>
 <code>VM=VM*translate (0,0,-2.5);</code>
 <code>viewMatrix (VM);</code>
 <code>pinta_escena ();</code></p> |

Solució: b)

7. (1 punt) Es vol realitzar una vista en planta (visió des de dalt) d'una escena/objecte que està centrat a l'origen amb una capsula contenidora de mides 10x10x10. Quina de les següents definicions et sembla correcta per definir la posició + orientació de la càmera (per a calcular la viewMatrix)? Sabem que la càmera és perspectiva i els angles de les rotacions estan en graus.

- a) $OBS = (0,10,0)$; $VRP = (0,0,0)$; $up = (0,1,0)$;
VM = lookAt (OBS, VRP, up);
viewMatrix(VM);
- b) $OBS = (0,0,0)$; $VRP = (0,10,0)$; $up = (0,0,-1)$;
VM = lookAt (OBS, VRP, up);
viewMatrix(VM);
- c) VM = translate (0,0,-10);
VM = VM * rotate (90, 1,0,0);
viewMatrix(VM);
- d) VM = translate (0,0,-10);
VM = VM * rotate (-90, 0,1,0);
viewMatrix(VM);

Solució: c)

8. (1 punt) Una aplicació permet, prement la tecla 'o', permutar entre una càmera perspectiva i una axonomètrica, ambdues amb el mateix ZNear i ZFar. Totes dues càmeres veuen l'escena completa i sense deformacions. Un estudiant dubte de quina càmera és la que està activa en un cert moment, quin dels següents experiments li aconsellaries fer per a deduir-ho?

- a) Movent l'observador en la direcció del VRP, la perspectiva retallarà per culpa del ZNear i l'axonomètrica no.
- b) Fent un resize de la finestra, la perspectiva deformarà i l'axonomètrica no.
- c) Movent l'observador en la direcció del VRP, la grandària de l'objecte no canviarà en l'axonomètrica, sí en la perspectiva.
- d) Girant la càmera en tercera persona (mitjançant angles d'Euler), l'axonomètrica retallarà i la perspectiva no.

Solució: c)