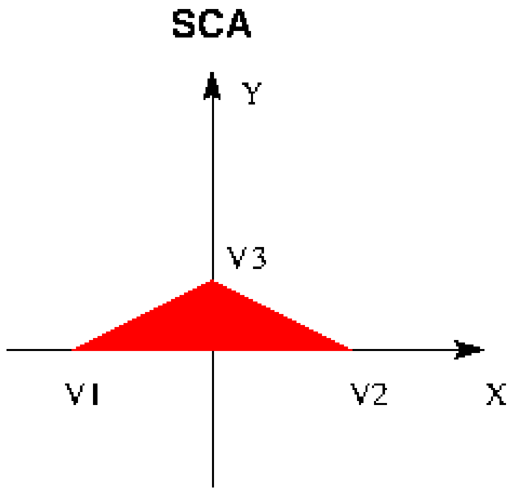
Examen parcial d'IDI 2015-2016, Q2. 20/4/2016 12:30

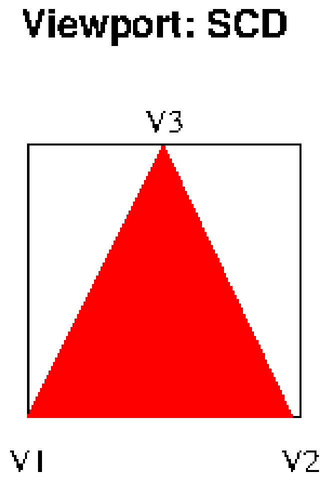
Nom i cognoms: Temps: 1h 20'

1. *(1.5 punts)* Tenim una escena amb un triangle vermell amb vèrtexs V1 = (-2, 0, 0), V2 = (2, 0, 0) i V3 = (0, 1, 0). Suposant que tenim un *viewport* quadrat de 600x600 píxels que ocupa tota la finestra gràfica.

   Indica TOTS els paràmetres d'una càmera ortogonal (axonomètrica) que permeti veure a la vista un triangle de manera que els seus vèrtexs en coordenades de dispositiu siguin: V1 = (0, 0), V2 = (600, 0) i V3 = (300, 600). Justifica la resposta.

   Solució:





Per a veure el triangle en el *viewport* com indica l'enunciat (part dreta de la figura), s'ha de mirar des de la part positiva de les Z, perquè la base del triangle en el *viewport* té el vèrtex V1 a l'esquerra i el vèrtex V2 a la dreta.

El VRP pot ser el centre de la capsa contenidora del triangle (punt que queda projectat al centre del *viewport*), és a dir podria ser el punt **VRP = (0, 0.5, 0)**, i per tant l'observador haurà d'estar en direcció perpendicular al triangle en la direcció de les Z positives, és a dir, per exemple en **OBS = (0, 0.5, 5)**. La vertical de la càmera és la vertical del triangle original, per tant el vector **up = (0, 1, 0)**.

Per als paràmetres de la càmera ortogonal cal donar el *window* (*left, right, bottom, top*), el ZNear i el ZFar. El *window*, que es dóna en coordenades de l'observador, ha d'anar des del vèrtex V1 fins al vèrtex V2 en l'eix de les X de l'observador, per tant ***left* = -2** i ***right* = 2**, i des de la base del triangle (aresta V1-V2) fins al vèrtex V3 en les Y de l'observador, per tant ***bottom* = -0.5** i ***top* = 0.5**. El ZNear i ZFar poden ser, per exemple, **ZNear = 5** i **ZFar = 6**, perquè tenim l'observador a distància 5 del triangle.

2. *(1.5 punts)* Es vol visualitzar una escena formada per dos objectes:

- Un prisma recte de base quadrada de costat 2 i alçada 5, que està situat amb el centre de la seva base al punt (0,0,0) i amb cares paral·leles als plans coordenats. Disposem del mètode `pintaCub()` que envia a pintar la geometria del model d'un cub de costat 1 centrat a l'origen de coordenades.
- Un Patricio d'alçada 2 que està situat al damunt de l'objecte anterior, és a dir, la base de la capsa del Patricio ha d'estar sobre la cara superior del prisma i mirant cap a les X+. Disposem del mètode `pintaPatricio()`, que pinta un Patricio que mira cap a les Z+ i la seva capsa contenidora està determinada pels punts: `(Patxmin, Patymin, Patzmin), (Patxmax, Patymax, Patzmax)`.

Suposant que la càmera està correctament inicialitzada, indica el pseudocodi d'una funció `pintaEscena()` que permeti visualitzar l'escena descrita especificant el codi (o pseudocodi) que permet trobar les TGs requerides per a cada objecte i utilitzant els mètodes `pintaCub()` i `pintaPatricio()`. Justifica la resposta.

Solució:

Per a pintar el prisma de base quadrada de 2x2 i alçada 5, cal escalar el model del cub segons els factors d'escala (2, 5, 2). Com que el cub que es pinta amb el mètode `pintaCub()` està centrat a l'origen, caldrà finalment traslladar-lo la meitat de l'alçada cap a les Y positives per a fer que la base estigui centrada en el punt (0, 0, 0).

Transformacions al cub: `TG = T(0, 2.5, 0) * S(2, 5, 2)`

Per a pintar el Patricio com es demana, el seu model s'ha d'escalar i girar, per tant primer podem fer que aquest estigui situat amb el centre de la seva base a l'origen de coordenades, és a dir primer de tot el traslladem segons el vector (-CentreBasePat.x, -CentreBasePat.y, -CentreBasePat.z), on CentreBasePat és el centre de la base de la capsa contenidora del Patricio: CentreBasePat = ((Patxmin+Patxmax)/2, Patymin, (Patzmin+Patzmax)/2).

Seguidament, com que ha de fer d'alçada 2 (escalat uniformement) cal escalar-lo en les tres coordenades pel factor: `escala = 2/(Patymax-Patymin)`. I com que ha d'estar mirant cap a les X+ i originalment està mirant cap a les Z+, cal també fer-li una rotació de 90 graus respecte de l'eix Y.

I finalment cal posicionar el centre de la base del Patricio al punt (0, 5, 0) que és el punt central de la cara superior del prisma, per tant caldrà traslladar-lo segons el vector (0, 5, 0).

Transformacions al Patricio:

```
TG = T(0, 5, 0) * Ry(90) * S(escala, escala, escala) * T(-CentreBasePat)
```

Així doncs, el codi de la funció `pintaEscena()` quedaria de la següent manera:

```
pintaEscena()
{
  gml::mat4 TG = Translació (0, 2.5, 0);
  TG = TG * Escalat (2, 5, 2);
  modelMatrix (TG);
  pintaCub();
  glm::vec3 CentreBasePat((Patxmin+Patxmax)/2, Patymin, (Patzmin+Patzmax)/2);
  float escala = 2/(Patymax-Patymin);
  TG = Translació (0, 5, 0);
  TG = TG * Rotació_y(90);
  TG = TG * Escalat (escala, escala, escala);
  TG = TG * Translació (-CentreBasePat);
  modelMatrix (TG);
  pintaPatricio();
}
```

Examen parcial d'IDI 2015-2016, Q2. 20/4/2016 12:30

Nom i cognoms: Temps: 1h 20'

3. *(1 punt)* Suposant que tenim la matriu de transformació de model (`TG`), la matriu de canvi de punt de vista (`view`) i la matriu de projecció (`proj`), quina és la multiplicació correcta que transforma el vèrtex (`vertex`) a coordenades de clipping? (el vèrtex ja està en coordenades homogènies, és a dir és un vec4(x,y,z,1)).

   a) `proj * TG * view * vertex`.
   b) `vertex * proj * view * TG`.
   c) `vertex * TG * view * proj`.
   d) `proj * view * TG * vertex`.

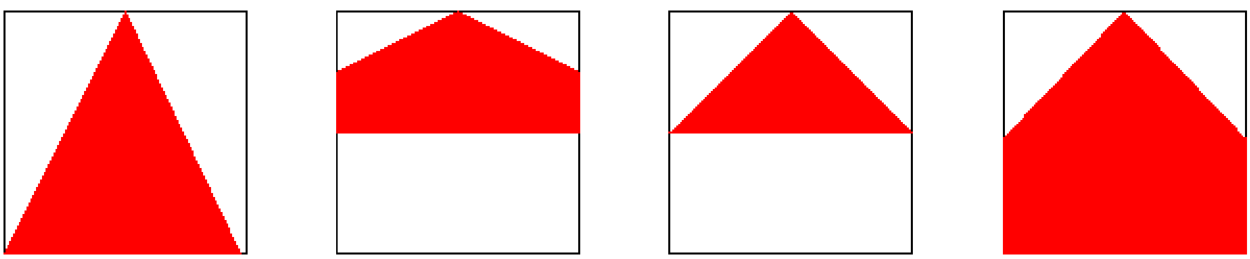
   Solució: d)

4. *(1 punt)* Un estudiant defineix la seva càmera amb OBS, d, v i up, essent d la distància entre OBS i VRP i v el vector normalitzat que va d'OBS a VRP, i calculant VRP com: **VRP = OBS + d * v**.
   En un cert moment, l'estudiant incrementa d i actualitza VRP i la view matrix però no la projection matrix, quin efecte tindrà en la visualització de l'escena?

   a) Com que no actualitza l'òptica, retallarà l'escena per Znear.
   b) Veurà l'escena més petita, el punt d'enfoc està més lluny.
   c) Veurà exactament el mateix.
   d) Afectarà en la deformació perspectiva que observarà.

   Solució: c)

5. *(1 punt)* Donada la descripció de l'escena de l'exercici 1 i havent inicialitzat les matrius de càmera (view) i projecció (proj) 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)

   Solució: b)

6. *(1 punt)* Imaginem que tenim l'escena de l'exercici 2, és a dir tenim les TGs necessàries per ubicar els dos objectes de forma correcta. Volem posicionar una càmera de manera que es vegi en el *viewport* el Patricio centrat i mirant cap a la càmera (de cara a la càmera). Quina d'aquestes inicialitzacions dels paràmetres de càmera seria correcta?

   a) VRP = (0, 6, 0), OBS = VRP + 2*radi_esfera*v, amb v=(0,1,0)), up = (1, 0, 0).

   b) VRP = ((Patxmin+Patxmax)/2, (Patymin+Patymax)/2, (Patzmin+Patzmax)/2),
      OBS = (VRP.x + 10, VRP.y, VRP.z), up = (0, 1, 0).

   c) VRP = (1, 6, 0), OBS = (15, 6, 0), up = (0, 1, 0).

   d) VRP = (0, 6, 0), OBS = (0, 6, 10), up = (0, 1, 0).

   Solució: c)

7. *(1 punt)* Tenim un objecte al que se li ha aplicat la següent transformació de model (TG) per a ubicar-lo a l'escena (C és el centre de l'objecte):

```
TG = Translació (3,0,3);
TG = TG*Rotacio_z (-90);
TG = TG*Rotacio_x (90);
TG = TG*Rotacio_y (90);
TG = TG*Escala (1/mida,1/mida,1/mida);
TG = TG*Translació (-C.x,-C.y,-C.z);
```

   Indica amb quina de les següents transformacions aconseguiríem l'objecte centrat al mateix punt i orientat de la mateixa manera però essent el doble de gran.

   a)
```
TG = Translació (3,0,3);
TG = TG*Rotacio_x (90);
TG = TG*Escala (2/mida,2/mida,2/mida);
TG = TG*Translació (-C.x,-C.y,-C.z);
```

   b)
```
TG = Translació (-3,0,-3);
TG = TG*Rotacio_z (90);
TG = TG*Rotacio_x (90);
TG = TG*Rotacio_y (-90);
TG = TG*Escala (2/mida,2/mida,2/mida);
TG = TG*Translació (-C.x,-C.y,-C.z);
```

   c)
```
TG = Translació (3,0,3);
TG = TG*Rotacio_z (-90);
TG = TG*Rotacio_x (90);
TG = TG*Rotacio_y (90);
TG = TG*Translació (-C.x,-C.y,-C.z);
TG = TG*Escala (2/mida,2/mida,2/mida);
```

   d) Cap de les altres és correcte.

   Solució: a)

Examen parcial d'IDI 2015-2016, Q2. 20/4/2016 12:30

Nom i cognoms: Temps: 1h 20'

8. *(1 punt)* Tenim una escena en la que utilitzem una càmera amb OBS = (-5,0,3), VRP = (5,0,3) i up = (0,0,1). Quin conjunt de transformacions geomètriques permetrien definir la mateixa càmera, és a dir, generar la mateixa viewMatrix? (no modifiquem l'òptica).

a)
```
VM = Translació(0,0,-10);
VM = VM*Rotacio_z (90);
VM = VM*Rotació_y (-90);
VM = VM*Translació (-5,0,-3)
ViewMatrix (VM);
```
b)
```
VM = Translació(0,0,-10);
VM = VM*Rotacio_z (90);
VM = VM*Rotació_y (90);
VM = VM*Translació (-5,0,-3)
ViewMatrix (VM);
```
c)
```
VM = Translació (-5,0,-3)
VM = VM*Rotació_y (90);
VM = VM*Rotacio_z (90);
VM = VM*Translació(0,0,-10);
ViewMatrix (VM);
```
d)
```
VM = Translació (-5,0,-3)
VM = VM*Rotació_y (-90);
VM = VM*Rotacio_z (90);
VM = VM*Translació(0,0,-10);
ViewMatrix (VM);
```

Solució: b)

9. *(1 punt)* La crida `TP = perspective (M_PI/4, 1, 3, 6)`, defineix la matriu de projecció d'una càmera perspectiva. La ra del *viewport* és 1. S'envia a pintar un cub d'aresta 2 centrat a l'origen i es veu sencer com un quadrat que ocupa tot el *viewport*. Si es modifica la ra del window i passa a ser 2 (i no es modifica cap altre paràmetre de la càmera), què es veurà quan tornem a pintar el cub?

a) Veurem un rectangle el doble d'alt que d'ample.
b) Com que ra del window és més gran que 1 es segueix veient un quadrat.
c) Veurem un rectangle el doble d'ample que d'alt.
d) Si no modifiquem el FOV el quadrat quedarà retallat.

Solució: a)