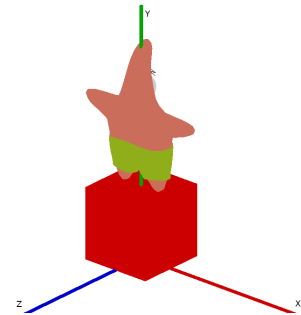


Normativa preguntes curtes

1. Responen 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 cub vermell de costat 10 amb el centre de la seva base a l'origen de coordenades i un Patricio d'alçada 20 ubicat amb el centre de la base de la seva capsa mínima contenidora al damunt del cub (centre de la base de la capsa del Patricio al punt $(0,10,0)$) i es troba mirant en direcció Z-. El Patricio està escalat uniformement.

Pots veure aquesta escena en la figura, a la que li hem afegit els eixos coordenats per a fer-la més clara.



1. (1 punt) Tenint en compte que volem pintar l'**Escena 1**, resol els següents apartats:

- a) Disposem del model del Patricio del que sabem que el centre de la seva capsa contenidora està en el punt **Pcentre**, que les mides de la capsa són **midaX**, **midaY** i **midaZ** i que el model mira inicialment en direcció Z+. Completa la funció següent que calcula i retorna la TG necessària per pintar el Patricio de l'escena.

```
glm::mat4 TG_Patricio()
{
    TG = I;

    ...

    return (TG);
}
```

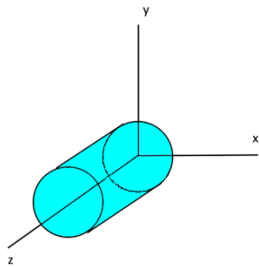
- b) Del model del cub sabem que si cridem a la funció **pintaCub()** ens pinta un cub vermell de costat 1 centrat a l'origen. Completa la funció següent que calcula i retorna la TG necessària per pintar el cub de l'escena.

```
glm::mat4 TG_Cub()
{
    TG = I;

    ...

    return (TG);
}
```

2. (1 punt) Tenim una escena amb un cilindre blau de base de radi 1, llargada 4, amb l'eix ubicat sobre l'eix Z+ i amb el centre de la base que té Z mínima situat a l'origen de coordenades. Completa els paràmetres d'una càmera ortogonal que permeti veure en pantalla un quadrat blau centrat al viewport i que ocupi la meitat d'aquest tant en vertical com en horitzontal. Tenim un viewport (vista) de 800x800.



```
VM = lookAt ( (2, 0, 2) , , );
PM = ortho ( , , , , , );
```

3. (1 punt) Tenim una escena amb un cub de costat 1 i cares paral·leles als plans coordenats amb el vèrtex de coordenades mínimes a l'origen de coordenades. Pintem aquesta escena amb una càmera ortogonal amb OBS=(3,0.5,0.5), VRP=(0.5,0.5,0.5), up=(0,1,0), Window=(-1,1,-0.5,0.5), ZNear=2, ZFar=3 (les matrius estan enviades al vertex shader correctament). Suposant que pintem aquesta imatge en un viewport de 800x800 píxels:

- a) Dibuixa què es veurà en el viewport si pintem el cub amb els següents vertex i fragment shaders (els colors els pots deixar indicats).

Vertex Shader:

```
in vec3 vertex;
uniform mat4 PM, VM;

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

Fragment Shader:

```
out vec4 FragColor;

void main () {
    if (gl_FragCoord.x>400 && gl_FragCoord.y>400)
        discard;
    else
        FragColor = vec4 (0, 0, 1, 1);
}
```

- b) En quin fragment (coordenades de dispositiu) es pintarà el vèrtex (1,1,1) del cub? Indica les coordenades x i y del fragment.

4. (1 punt) Tenim un dibuix que utilitza els colors C1=verd, C2=blau, C3=vermell i C4=groc, tots ells de màxima intensitat i usats en la mateixa quantitat dins del dibuix.
- a) Tenim una impressora CMY que està carregada amb tintes CMY de manera que el Cian (C) està carregat al 100% i les altres dues al 50%. Si s'imprimeix aquest dibuix repetidament, quina de les tintes de la impressora s'acabarà abans?
- b) Si a la impressora li falla la tinta groga (Y) 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 i C4 un cop impressos en paper blanc?
- c) Suposant que el dibuix està ben imprès sobre paper blanc, indica de quin color es veuran les parts del dibuix dels colors C3 i C4 si ens mirem el dibuix a través d'un filtre cian (només deixa passar la llum cian).

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) Un nen dibuixa un sol de color vermell, i en pantalla el veu perfectament d'aquest color. Si quan l'envia a imprimir el sol es veu de color blau en el paper imprès, què ha passat?
- a) El paper era magenta i la impressora funciona perfectament (té totes les tintes).
 - b) El paper era blanc però la impressora no té tintes cian i groga.
 - c) El paper era cian i la impressora s'ha quedat sense tinta groga.
 - d) No és possible que el dibuix es vegi imprès de color blau.
6. (0.5 punts) Un VAO (Vertex Array Object) ens serveix per:
- a) Encapsular les dades d'un o més objectes.
 - b) Encapsular les dades d'un model i les seves TGs.
 - c) Encapsular les dades d'un model.
 - d) Encapsular l'array de vèrtexs d'un objecte.
7. (0.5 punts) En què consisteix l'algoritme de visualització anomenat com a projectiu?
- a) S'ordenen els triangles de més proper a més llunyà respecte a l'observador, i per aquells que són més propers, es determina en quins píxels es projecten, i aquests s'actualitzen amb el color del triangle que és més proper.
 - b) Per a cada triangle es determina en quins píxels es projecta, i en cada píxel que es projecta, si queda per davant del que s'havia projectat anteriorment, s'actualitza el color del píxel amb el seu color.
 - c) Per a cada píxel es determina quins triangles es projecten en ell, i s'actualitza el color del píxel amb el color del més proper a l'observador.
 - d) Per a cada triangle es determina en quins píxels es projecta, i en cada píxel que es projecta s'actualitza aquest amb el seu color.

8. (0.5 punts) Una càmera perspectiva està posicionada mitjançant angles d'Euler amb el codi següent:

```
VM = Translate (0, 0, -10);  
VM = VM * Rotate (-45, (0, 0, 1));  
VM = VM * Rotate (-90, (0, 1, 0));  
VM = VM * Translate (-5, -5, 0);
```

Quin dels següents conjunts de paràmetres permetrà tenir la mateixa viewMatrix posicionada fent servir ara OBS, VRP i Up? (tots els angles estan en graus sexagesimals).

- a) OBS = (10, 0, 0); VRP = (0, 0, 0); Up = (0, 1, 1)
- b) OBS = (15, 5, 0); VRP = (5, 5, 0); Up = (0, 1, -1)
- c) OBS = (15, 5, 0); VRP = (5, 5, 0); Up = (0, 1, 1)
- d) OBS = (15, 0, 0); VRP = (0, 0, 0); Up = (0, 0, 1)

9. (0.5 punts) Es disposa d'una funció `pintaCub()` que envia a pintar el VAO d'un cub d'aresta 10 amb el centre del cub a l'origen de coordenades. Tenim una càmera definida amb OBS= (0,0,20), VRP=(0,0,0), up=(0,1,0) Quin dels següents codis és correcte per visualitzar una escena amb 4 cubs que quan es pinten formen una T al window amb el cub central superior de la T centrat al centre del window. Nota: el vertex shader està correctament definit i rep com uniform una matriu TG de 4x4; l'òptica és ortogonal i està definida també correctament.

- a)

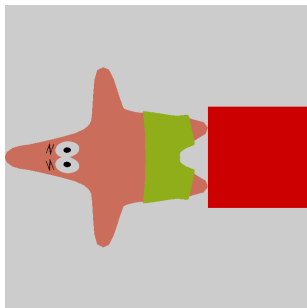
```
TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (0,-10,0); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (10,0,0); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (-10,0,0); modelMatrix (TG);  
pintaCub();
```
- b)

```
TG1=I; TG1= TG1*Translació (0,0,10); modelMatrix (TG1);  
TG2=I; TG2= TG2*Translació (0,-10,0); modelMatrix (TG2);  
TG3=I; TG3= TG3*Translació (10,0,0); modelMatrix (TG3);  
TG3=I; TG3= TG3*Translació (-10,0,0); modelMatrix (TG3);  
pintaCub();  
pintaCub();  
pintaCub();  
pintaCub();
```
- c)

```
pintaCub();  
TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (0,-10,0); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (10,0,0); modelMatrix (TG);  
pintaCub();  
TG=I; TG= TG*Translació (-10,0,0); modelMatrix (TG);
```
- d)

```
TG=I; TG= TG*Translació (0,0,10); modelMatrix (TG);  
pintaCub();  
TG= TG*Translació (0,-10,0); modelMatrix (TG);  
pintaCub();  
TG= TG*Translació (10,0,0); modelMatrix (TG);  
pintaCub();  
TG= TG*Translació (-10,0,0); modelMatrix (TG);  
pintaCub();
```

10. (0.5 punts) Definida correctament una càmera perspectiva per veure tota una escena, es permet a l'usuari que via la interfície modifiqui la distància de VRP a OBS, fent que el VRP s'allunyi d'OBS en la direcció de visió (OBS no varia). Què podrem observar?
- Tindrem un efecte semblant a un zoom out en successives visualitzacions.
 - Si incrementem molt la distància, per veure tota l'escena ocupant el màxim del viewport s'hauria de modificar també l'angle d'obertura de la càmera.
 - Cal limitar la distància VRP-OBS a què com a molt sigui igual a Zfar per a que funcioni el pipeline de visualització correctament.
 - Cap de les altres és correcta.
11. (0.5 punts) En una càmera perspectiva, quan l'usuari redimensiona el viewport, què cal fer per evitar la deformació de l'escena?
- Cal igualar la ra del window a la del viewport.
 - Si la ra del viewport és menor que 1 cal modificar també l'angle FOV.
 - Si la ra del viewport és major que 1 no cal fer res.
 - Totes les altres són correctes.
12. (0.5 punts) Quin és el procés del Pipeline de visualització d'OpenGL que permet passar de vèrtexs en Sistema de Coordenades de Clipping (SCC) a vèrtexs en Sistema de Coordenades Normalitzades (SCN)?
- Divisió de perspectiva
 - Rasterització
 - Clipping (retallat)
 - Transformació Món-Dispositiu
13. (0.5 punts) Tenint en compte l'**Escena 1**, indica quins dels paràmetres de posició i orientació de la càmera ens permetran veure la imatge de la figura. L'òptica està ben definida.



- $OBS = (0, 15, -25)$; $VRP = (0, 15, 10)$; $Up = (-1, 0, 0)$.
 - $OBS = (10, 15, 0)$; $VRP = (0, 15, 10)$; $Up = (1, 0, 0)$.
 - $OBS = (0, 15, -20)$; $VRP = (0, 15, 0)$; $Up = (1, 0, 0)$.
 - $OBS = (0, 15, 20)$; $VRP = (0, 15, 0)$; $Up = (-1, 0, 0)$.
14. (0.5 punts) Tenim una escena que la seva capsula mínima contenidora està definida pels punts $P_{min}=(-5,-5,-5)$ i $P_{max}=(5,5,5)$. Quin dels següents conjunts de paràmetres d'una òptica perspectiva permetrà veure tota l'escena sense deformar ni retallar amb una càmera posicionada amb $OBS=(10,0,0)$, $VRP=(0,0,0)$, $Up=(0,1,0)$ i en un viewport de 800x400. (Els angles estan en graus).
- $FOV = 90$; $ra = 0.5$; $Z_n = 4$; $Z_f = 20$;
 - $FOV = 60$; $ra = 1$; $Z_n = 5$; $Z_f = 10$;
 - $FOV = 60$; $ra = 2$; $Z_n = 5$; $Z_f = 15$;
 - $FOV = 90$; $ra = 2$; $Z_n = 5$; $Z_f = 15$;

15. (0.5 punts) Quina de les següents opcions NO és certa quan parlem de la consistència dins d'una família de productes?
- a) Transmet marca.
 - b) Facilita l'adopció del producte.
 - c) Només és necessària a nivell d'estructures visibles per transmetre *look&feel*.
 - d) Millora l'aprenentatge per a usuaris familiaritzats amb altres productes de la mateixa família.
16. (0.5 punts) Si volem dissenyar una pàgina web, quina de les següents accions NO és aconsellable en cap cas?
- a) Podem utilitzar colors amb significat però mai com a única informació decisiva.
 - b) El text que hi posem ha de tenir una mida adient per a la lectura fàcil.
 - c) Hem de posar molta informació per a què es vegi tot al primer cop d'ull.
 - d) El text que hi posem ha de contrastar bé amb el fons.