

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: Tenim una escena formada per: un terra modelat per un quadrat ubicat en el pla $Y=0$ de 5×5 centrat a l'origen i costats paral·lels als eixos X i Z de coordenades, un Patricio d'alçada 1 amb el centre de la base de la seva capsula contenidora al punt $(0,0,0)$ mirant cap a $Z+$ i un segon Patricio d'alçada 2 amb el centre de la base de la seva capsula a $(1.5, 0, 0)$ i mirant també cap a $Z+$. Els dos Patricios estan escalats uniformement.

1. (1 punt) Indica TOTS els paràmetres d'una càmera perspectiva que permeti veure l'**Escena 1** centrada, sense retallar i de manera que la seva esfera contenidora ocupi el màxim del viewport (càmera en tercera persona). Els dos Patricios s'han de veure de cara amb una direcció de visió paral·lela a l'eix Z . El viewport és quadrat. (No cal calculadora, podeu deixar valors indicats)

Possible solució:

puntMinim capsula= $(-2.5, 0, -2.5)$; puntMaxim capsula= $(2.5, 2, 2.5)$;

VRP = $(0, 1, 0)$; centre de la capsula contenidora

$R = \text{sqrt}(2.5^2 + 1 + 2.5^2) = \text{sqrt}(6.25 + 1 + 6.25) = 3.67$;

$d = 2 * R$;

OBS = $(0,1,0) + d * (0,0,1)$; mira en direcció paral·lela a Z

up = $(0,1,0)$;

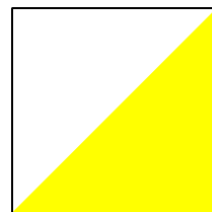
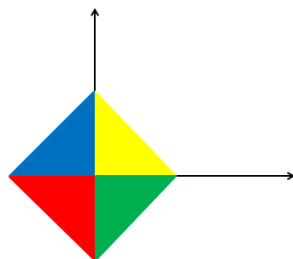
FOV = $2 * \text{asin}(R/d) = M_PI/3$;

ra = 1; mateixa ra que el viewport que és quadrat

ZNear = $d - R$; **ZFar** = $d + R$;

2. (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.

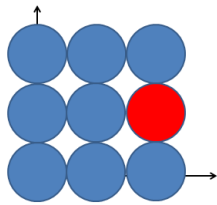
Completa els paràmetres necessaris d'una càmera ortogonal per a què al viewport, que és quadrat, es vegi el que teniu a la imatge de la dreta.



VM = **lookAt** ($(2,2,2)$, $(2,2,0)$, $(1,0,0)$);
PM = **ortho** (-2 , 2 , -2 , 2 , 1 , 3);

3. (1 punt) Disposem d'una rutina `pintaEsfera()` que pinta una esfera de radi 5 centrada a l'origen. A partir d'aquesta rutina es vol construir una escena com la de la imatge de la figura on cada esfera té radi 0.5 i totes les esferes estan centrades en el pla XY.

Completa el càlcul de la TG que cal per a poder pintar l'esfera vermella tenint en compte que cal pintar-la usant la rutina `pintaEsfera()`.



```
tg_vermella () {
    TG = I;
    ...
    return (TG);
}
```

```
tg_vermella () {
    TG = I;
    TG = TG * translació (2,1,0);
    TG = TG * escala (0.1,0.1,0.1);
    return (TG);
}
```

4. (1 punt) Dibuixem un quadrat mitjançant dos triangles amb vèrtexs $V1=(-1,-1,0)$, $V2=(1,-1,0)$, $V3=(-1,1,0)$, $V4=(1,1,0)$, $V5=(1,-1,0)$ i $V6=(1,1,0)$. Suposant que el viewport és de 600x600, dibuixa què es veurà en el viewport si pintem el quadrat amb els següents vertex i fragment shaders (els colors els pots deixar indicats).

Vertex Shader:

```
in vec3 vertex;
```

```
void main()
```

```
{
```

```
    gl_Position = vec4(vertex,1);
```

```
}
```

Fragment Shader:

```
out vec4 FragColor;
```

```
void main ()
```

```
{
```

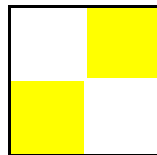
```
    if ((gl_FragCoord.x>300 && gl_FragCoord.y>300) ||
        (gl_FragCoord.x<300 && gl_FragCoord.y<300))
```

```
        FragColor = vec4 (1, 1, 0, 1);
```

```
    else
```

```
        FragColor = vec4 (1, 1, 1, 1);
```

```
}
```



5. (1 punt) Quin és el nombre de VAOs i VBOs que hem de crear per poder pintar l'**Escena 1** en els dos casos següents?

- a) Si per al model de cada objecte volem guardar només les coordenades dels vèrtexs

VAOs = 2 ;

VBOs = 2 ;

- b) Si per al model de cada objecte volem guardar les coordenades dels vèrtexs i el color dels vèrtexs

VAOs = 2 ;

VBOs = 4 ;

6. (1 punt) Tenim una escena amb un cub de costat 1 i cares paral·leles als plans coordenats amb el vèrtex mínim a l'origen de coordenades. Pintem aquesta escena amb una càmera ortogonal amb $OBS=(0,10,0)$, $VRP=(0,0,0)$, $up=(0,0,-1)$, $Window=(-2,2,-1,1)$, $ZNear=5$, $ZFar=15$. Suposant que pintem aquesta imatge en un viewport de 800x600 píxels, en quin fragment (coordenades de dispositiu) es pintarà el vèrtex (1,1,1) del cub? Indica les coordenades x i y del fragment.

Solució: Fragment: (600, 0)

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
7				
8				
9				
10				

Num	A	B	C	D
11				
12				
13				
14				

7. (0.5 punts) Quina de les següents afirmacions és certa:

- a) Quan tenim el vèrtex en coordenades de model, si el multipliquem per VM obtenim les coordenades de clipping.
- b) Quan tenim el vèrtex en coordenades de model, si el multipliquem per VM*TG obtenim les coordenades d'observador.
- c) Quan tenim el vèrtex en coordenades d'observador, si el multipliquem per VM obtenim les coordenades de clipping.
- d) Quan tenim el vèrtex en coordenades d'aplicació, si el multipliquem per VM*PM obtenim les coordenades de clipping.

8. (0.5 punts) Quina de les següents accions NO s'efectua en el procés de Visualització d'OpenGL?

- a) Passar de Coordenades de Model a Coordenades d'Aplicació.
- b) Passar de Coordenades de Dispositiu a Coordenades de Clipping.
- c) Passar vèrtex de Sistema de Coordenades de Clipping a Sistema de Coordenades Normalitzades.
- d) Passar vèrtex de Sistema de Coordenades d'Aplicació a Sistema de Coordenades d'Observador.

9. (0.5 punts) Suposem una càmera amb paràmetres OBS=(10,6,0); VRP=(0,6,0); i up=(0,1,1) per veure l'**Escena 1**. Quina de les següents transformacions geomètriques aconseguirà la mateixa View Matrix (VM) que aconseguiria la crida lookAt(OBS,VRP,up)?

- a) $VM = T(0,0,-10) * R_z(90) * R_x(45) * T(0,-6,0)$
- b) $VM = T(0,-6,0) * R_y(-90) * R_z(-45) * T(0,0,-10)$
- c) $VM = T(0,0,-10) * R_z(-45) * R_y(-90) * T(0,-6,0)$
- d) $VM = T(0,-6,0) * R_y(-90) * T(0,0,-10)$

10. (0.5 punts) Suposem que volem afegir un Legoman a l'**Escena 1** descrita al full anterior. Aquest Legoman ha de tenir el centre de la base de la seva capsula contenidora situat al punt (0, 0, 1.5), ha d'estar mirant cap a Z- i ha de fer alçada 3. Sabent que inicialment el Legoman està mirant cap a Z+, i que la caixa 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 (0,0,1.5);
TG = TG * Rotate (180, 0, 1, 0);
TG = TG * Scale (3/(Lymax-Lymin), 3/(Lymax-Lymin), 3/(Lymax-Lymin));
TG = TG * Translate (-(Lxmax+Lxmin)/2, -Lymin, -(Lzmax+Lzmin)/2);
```
- b)

```
TG = Translate (0,0,1.5);
TG = TG * Rotate (180, 0, 1, 0);
TG = TG * Scale ((Lymax-Lymin)/3, (Lymax-Lymin)/3, (Lymax-Lymin)/3);
TG = TG * Translate (-(Lxmin+Lxmax)/2, -Lymin, -(Lzmin+Lzmax)/2);
```
- c)

```
TG = Translate (-(Lxmin+Lxmax)/2, -Lymin, -(Lzmin+Lzmax)/2);
TG = TG * Scale (3/(Lymax-Lymin), 3/(Lymax-Lymin), 3/(Lymax-Lymin));
TG = TG * Rotate (180, 0, 1, 0);
TG = TG * Translate (0,0,1.5);
```
- d)

```
TG = Translate (-(Lxmax-Lxmin)/2, -Lymin, -(Lzmax-Lzmin)/2);
TG = TG * Scale (3, 3, 3);
TG = TG * Rotate (180, 0, 1, 0);
TG = TG * Translate (0,0,1.5);
```

11. (0.5 punts) Indica quin és l'ordre correcte dels següents processos del Procés de Visualització:

1. Clipping (Retallat)
2. Transformació de coordenades de model (SCM) a coordenades d'aplicació (SCA)
3. Transformació de coordenades d'observador (SCO) a coordenades de clipping (SCC)
4. Rasterització
5. Transformació Món-Dispositiu (pas de SCN a SCD)

- a) 2 - 3 - 1 - 5 - 4
- b) 3 - 1 - 2 - 5 - 4
- c) 3 - 2 - 1 - 4 - 5
- d) 2 - 3 - 4 - 1 - 5

12. (0.5 punts) Del següent llistat d'afirmacions relatives a l'ús de colors en una interfície gràfica, indica quina és **FALSA**:

- a) Els colors saturats poden provocar fatiga visual.
- b) Per a que es vegi bé un text, és recomanable utilitzar contrast, per exemple fons clar per al text, i fons fosc per al fons.
- c) Si tenim botons del mateix color, sobre un fons que canvia d'intensitat, l'usuari els veurà diferents.
- d) És millor seleccionar molts colors diferents per a les icones d'una interfície, per a que sembli més senzill distingir els elements.

13. (0.5 punts) Per a què s'utilitza la inconsistència induïda *Induced inconsistency*?

- a) Per a que programes d'una mateixa empresa no resultin massa similars i resultin poc atractius a l'usuari.
- b) Per ressaltar, en una nova versió del programa, objectes que fan quelcom diferent a l'esperat.
- c) Per provar nous modes d'interacció.
- d) Per a que resulti més atractiva per a l'usuari una nova versió d'un programa.

14. (0.5 punts) Respecte al disseny de missatges d'error en un programa, indica l'afirmació que **NO és correcta**:

- a) Els missatges d'error han de donar un missatge clar a l'usuari i no un codi.
- b) Els missatges d'error han d'indicar a l'usuari com actuar.
- c) Els missatges d'error han de mostrar-se en pantalla durant un temps molt limitat per no molestar a l'usuari.
- d) Els missatges d'error han d'indicar què és el que està malament.