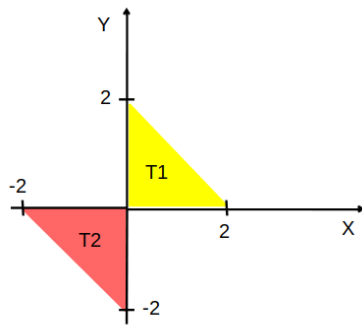


Normativa preguntes curtes

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.

1. (1 punt) Tenim un triangle amb vèrtexs $V1=(1,0,0)$, $V2=(0,1,0)$ i $V3=(-1,0,0)$ que es pinta amb el mètode `pintaTriangle()`. Usant aquest triangle volem pintar una escena com la que veieu a la figura, amb els dos triangles T1 i T2 tal i com es descriuen en la imatge.

Completa el càlcul de la transformació TG2 que cal per poder pintar el triangle T2.



```
tg_triangle_2 () {
    TG2 = I;

    ...

    return (TG2);
}
```

Solució:

```
tg_triangle_2 () {
    TG2 = I;
    TG2 = TG2 * Rotate (-45, (0, 0, 1));
    TG2 = TG2 * Scale (2/sqrt(2), 2/sqrt(2), 2/sqrt(2));
    TG2 = TG2 * Translate (0, -1 0);
    return (TG2);
}
```

2. (1 punt) Tenim una escena formada per un cub de costat 2 centrat a l'origen. Definim una càmera ortogonal amb paràmetres: $OBS = (4,1,-1)$, $VRP = (1,1,-1)$, $up = (0,1,0)$, $left = -2$, $right = 2$, $bottom = -2$, $top = 2$.

- a) Quins són els valors de ZNear i ZFar necessaris per a què el cub quedi completament dins del volum de visió i de manera ajustada (volum mínim)?

Solució: ZNear = 3; ZFar = 5.

- b) Si tenim un viewport de 800x600 (amplada x alçada), quines coordenades té el vèrtex del cub $V = (1,1,1)$ en els següents sistemes de coordenades?

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

Solució: $V(SCO) = (-2,0,-3)$; $V(SCN) = (-1,0,1)$; $V(SCD) = (0,300)$.

3. (1 punt) Dibuixem un quadrat amb vèrtexs $V1=(-1,-1,0)$, $V2=(1,-1,0)$, $V3=(1,1,0)$ i $V4=(-1,1,0)$. 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:

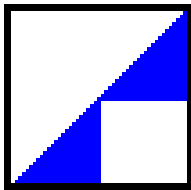
```
in vec3 vertex;

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.y > gl_FragCoord.x))
        discard;
    else
        FragColor = vec4 (0, 0, 1, 1);
}
```



4. (1 punt) Tinc un dibuix de color groc amb codificació $RGB = (0.7, 0.7, 0)$:

- a) Si imprimim un dibuix d'aquest color en una impressora CMY i usant un paper de color magenta al 70% (paper té 0.7 de magenta), de quin color veurem el dibuix? Indica la seva codificació RGB.

Solució: $RGB = (0.7, 0, 0)$

- b) Indica la codificació en format HSB del color del dibuix inicial.

H =

S =

B =

Solució: $HSB = (60, 1, 0.7)$

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 estudiant ha enviat a imprimir a una impressora CMY un dibuix de color verd, però quan el recull de la impressora el que hi veu en el paper és un dibuix de color cian. Què ha passat?
- a) La impressora no té tinta magenta.
 - b) El paper en què ha imprès és de color groc.
 - c) Imprimint aquest dibuix és impossible que passi això.
 - d) Cap de les altres respostes.
6. (0.5 punts) Suposem que tenim una escena centrada en el punt (5,10,0) i que la seva capsula contenedora té mides 10, 20, 10 en X, Y i Z respectivament. L'escena està pintada en un viewport quadrat usant una càmera inicialitzada amb les matrius que descriu el següent codi (els angles són en graus):

```
PM = ortho (-10, 10, -10, 10, 5, 15);
projectMatrix (PM);
VM = translate (0, 0, -10);
VM = VM*rotate (90, (1, 0, 0));
VM = VM*rotate (-90, (0, 1, 0));
VM = VM*translate (-5, -10, 0);
viewMatrix (VM);
```

Quins paràmetres OBS, VRP i up definirien exactament la mateixa View Matrix?

- a) OBS = (-5, 10, 0); VRP = (5, 10, 0); up = (0, -1, 0);
- b) OBS = (5, 10, 10); VRP = (5, 10, 0); up = (0, 1, 0);
- c) OBS = (-5, 10, -10); VRP = (0, 10, 0); up = (1, 0, 0);
- d) OBS = (5, 20, 0); VRP = (5, 10, 0); up = (-1, 0, 0);

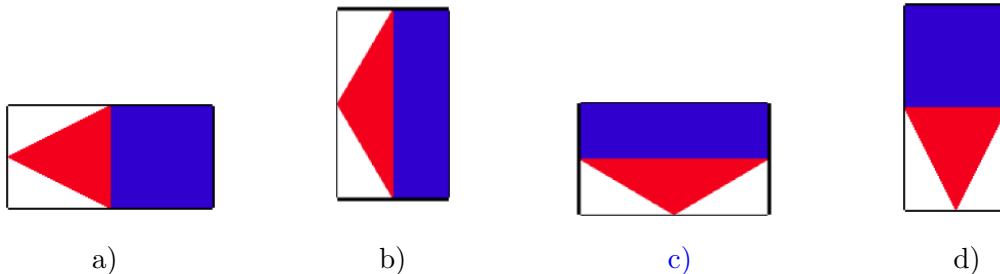
7. (0.5 punts) Una escena es defineix amb una esfera contenidora de radi R i centrada a l'origen. Tenim un viewport de 600×300 (amplada x alçada) i volem calcular els paràmetres per a una càmera perspectiva en tercera persona (escena centrada al viewport, ocupant el màxim i sense deformar). Si tenim l'observador situat a una distància d del centre de l'escena, quina de les següents fórmules és la correcta per calcular el FOV (Field Of View)?

- a) $FOV = \text{asin}(R/d)$;
- b) $hw = 2 * R$; $zn = R$; $FOV = \text{atan}(hw/(2*zn))$;
- c) $FOV = 2 * \text{asin}(R/d)$;
- d) $FOV = 2 * \text{atan}(R/d)$;

8. (0.5 punts) Tenim una piràmide de base quadrada de costat 2, amb la seva 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 és de 600×300 (amplada x alçada). Es fa la següent inicialització d'una càmera ortogonal (posició+orientació i òptica):

```
PM=ortho(-1, 1, -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);
pinta_escena ();
```

Quina de les següents imatges es veurà al viewport? Nota: Tots els angles estan en graus.



9. (0.5 punts) Disposem d'un model d'un ninot que la seva vertical està orientada com l'eix de les Y^+ (és a dir que el vector que uneix els peus amb el cap del ninot és paral·lel al vector $(0,1,0)$). El model del ninot, a més, mira en la direcció de l'eix de les Z^+ i està posicionat amb el centre de la base de la seva capsa contenidora al punt $(0,0,0)$. Si li apliquem la transformació geomètrica definida per el producte següent (els angles són en graus)

$$TG = \text{Translate}(1, 0, 0) * \text{Rotate}_Y(90) * \text{Rotate}_X(-90) * \text{Translate}(-1, 0, 0)$$

el ninot mirarà (en el sistema de coordenades de l'aplicació) en la direcció del vector:

- a) $(0,1,0)$
- b) $(0,-1,0)$
- c) $(1,0,0)$
- d) $(1,1,0)$

10. (0.5 punts) Suposem que tenim una escena en què la seva esfera contenidora està centrada a l'origen i té com a radi 50. Inicialment tenim la següent definició de càmera: FOV=60 (en graus), ra=1, ZNear=50, ZFar=150 i la viewMatrix s'inicialitza amb:

```
VM = I;  
VM = VM * Translate (0,0,-100);  
viewMatrix (VM);
```

Quins paràmetres ens caldrà canviar d'aquesta definició de càmera si volem tenir un efecte de zoom apropant l'observador cap al centre de l'esfera?

- a) FOV=30
 - b) VM = VM * Translate (0,0,-75) i ZNear=25
 - c) No es pot calcular la viewMatrix només amb una translació.
 - d) ZNear=25
11. (0.5 punts) El següent símbol que podem trobar a la carretera indicant que hi ha un hotel o hostel, de quin tipus és?



- a) Simbòlic
 - b) Arbitrari
 - c) Exemple
 - d) Similaritat
12. (0.5 punts) Dos estudiants estan comentant què cal modificar de l'òptica d'una càmera en tercera persona quan es fa el resize en òptica ortogonal i perspectiva per tal que no hi hagi deformació ni retallat de l'escena. El que diuen respectivament és:
- Estudiant A: *En les dues òptiques cal modificar el window modificant els paràmetres adjacents en cada òptica.*
- Estudiant B: *En l'òptica ortogonal cal modificar el window i en la perspectiva raw i segons el valor de raw també el FOV.*
- a) Cap dels dos estudiants té raó.
 - b) Tots dos estudiants tenen raó.
 - c) Només l'estudiant B ho farà correctament.
 - d) Només l'estudiant A ho farà correctament.
13. (0.5 punts) En termes d'usabilitat, una aplicació és eficient si...

- a) Donant el mínim input per part de l'usuari s'aconsegueix completar el màxim nombre de tasques.
- b) Té en compte estalviar treball per a l'equip d'IT (tècnics informàtics) que desenvolupen l'aplicació.
- c) Cap de les altres respostes és correcta.
- d) Permet a l'usuari completar les tasques de forma correcta.

14. (0.5 punts) Indica el nom del principi universal de disseny que tracta de prevenir els errors i facilitar que l'usuari no s'equivoqui, per exemple, detectant errors quan l'usuari introdueix dades en un formulari:
- a) Chunking
 - b) Rule of thirds
 - c) Garbage In - Garbage Out
 - d) LATCH
15. (0.5 punts) En els principis universals de disseny, el concepte de *chunking* està relacionat amb:
- a) La recomanació d'organitzar la informació de forma jeràrquica i agrupant els elements aproximadament de 5 en 5.
 - b) La capacitat que té l'usuari de retenir aproximadament 5 elements en la memòria temporal.
 - c) La capacitat dels usuaris de trobar informació en llistats de pocs elements.
 - d) La recomanació de mostrar a l'usuari pocs elements en pantalla.
16. (0.5 punts) Una plataforma té consistència si...
- a) Es manté constantment a l'usuari informat amb missatges d'error.
 - b) Tota la informació no està visible per a l'usuari.
 - c) S'utilitza el mateix hue per al text i el fons però amb diferent saturació.
 - d) Es manté una apariència específica en tota la gama dels seus productes.