

Normativa preguntes curtes

1. Responen les següents sis preguntes en el mateix full de l'enunciat.
2. Cal que les respostes siguin **clares**.
3. No es poden usar apunts ni calculadores ni cap dispositiu electrònic.

1. (1 punt) Tenim un dibuix format per dues franges (F1 i F2) de colors groc (F1) i magenta (F2).
- a) De quins colors es veuran les dues franges impreses usant una impressora CMY (la impressora funciona correctament) en els següents casos?
- a1) si el paper és de color cian:
- Solució:** La franja F1 es veurà verda i la franja F2 es veurà blava.
- a2) si el paper és de color verd:
- Solució:** La franja F1 es veurà verda i la franja F2 es veurà negra.
- b) Quina codificació tenen els colors originals de les franges (F1 i F2), groc i magenta, en HSB si tenim en compte que són colors purs i d'intensitat 0.8?

Solució: F1 (groc) serà: (60, 1, 0.8). F2 (magenta) serà: (300, 1, 0.8).

2. (1 punt) Volem muntar una escena posant un gerro sobre una taula. La taula no necessita cap transformació geomètrica per a situar-la a l'escena, però el gerro el volem uniformement escalat al doble de gran del que és originalment i situat amb la seva base centrada al punt (10, 5, 0).

Tenint en compte que els punts mínim i màxim de la capsa contenidora del model del gerro són: (xmin, ymin, zmin) i (xmax, ymax, zmax), completa la funció següent que calcula i retorna la TG necessària per a pintar aquest gerro a la nostra escena:

Solució:

```
tg_gerro () {  
    TG = I;  
    ...  
    return (TG);  
}  
  
tg_gerro () {  
    TG = I;  
    TG = TG * translate (10, 5, 0);  
    TG = TG * scale (2, 2, 2);  
    glm::vec3 cbase=((xmin+xmax)/2,ymin,(zmin+zmax)/2);  
    TG = TG * translate (-cbase);  
    return (TG);  
}
```

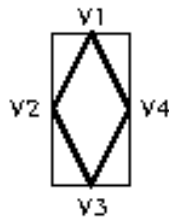
3. (1 punt) En la següent llista teniu etapes/tasques del procés de visualització d'OpenGL ordenades per ordre alfabètic. Escriu-les de nou refent l'ordre per a què sigui l'ordre en què es realitzen en el procés de visualització d'OpenGL.

Solució:

Clipping	Pas de coordenades del vèrtex a SCO
Fragment shader	Clipping
Pas de coordenades del vèrtex a SCO	Rasterització
Rasterització	Fragment shader

4. (1 punt) Pintem un quadrilàter amb vèrtexs $V1=(3,0,0)$, $V2=(0,0,-2)$, $V3=(-3,0,0)$ i $V4=(0,0,2)$.

a) Acaba d'omplir els paràmetres que falten d'una càmera ortogonal que en un viewport de 400x800 mostra la imatge següent:



```
VM = lookAt ((0,10,0), (0,0,0),  
PM = ortho ( , , , , 8, 12);
```

Solució:

```
VM = lookAt ((0,10,0), (0,0,0), (1,0,0);  
PM = ortho (-2, 2, -3, 3, 8, 12);
```

b) Quina és la relació d'aspecte del window (raw) de la càmera?

Solució: raw = 2/3

5. (1 punt) Una escena formada per tres objectes es pinta usant el mètode següent:

```
1- pinta_Escena () {  
2-     modelMatrix (TG1);  
3-     pinta_obj1 ();  
4-     modelMatrix (TG2);  
5-     pinta_obj2 ();  
6-     modelMatrix (TG3);  
7-     pinta_obj3 ();  
8- }
```

Suposant que TG1, TG2 i TG3 són variables globals que contenen les matrius de transformació que necessita cada objecte respectivament i que la crida modelMatrix(TG) només envia la matriu a la tarja gràfica, què afegiries i on en aquesta rutina pinta_Escena() per a que, donat un cert vector de translació (tx,ty,tz), en coordenades de l'aplicació, tota l'escena es traslladi segons aquest vector de translació?

Solució:

```
1- pinta_Escena () {  
    TGaux = translate (tx, ty, tz);  
    TG = TGaux * TG1;  
2- modelMatrix (TG);  
3- pinta_obj1 ();  
    TG = TGaux * TG2;  
4- modelMatrix (TG);  
5- pinta_obj2 ();  
    TG = TGaux * TG3;  
6- modelMatrix (TG);  
7- pinta_obj3 ();  
8- }
```

6. (0.5 punts) Lliga els principis i conceptes de les dues llistes següents:

- a- Nombre màgic 5 d'agrupament d'informació
- b- Organització alfabètica de la informació
- c- Representació similar a objecte o acció
- d- Completar figures regulars
- e- Percepció figures simples

- 1- Llei de tancament de Gestalt
- 2- LATCH
- 3- Llei de Pragnänz (bona figura)
- 4- Chunking
- 5- Representació icònica

Solució: a - 4; b - 2; c - 5; d - 1; e - 3;

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				X
8			X	
9				X

Num	A	B	C	D
10	X			
11			X	
12	X			

Num	A	B	C	D
13	X			
14				X
15			X	

7. (0.5 punts) Donada una càmera perspectiva en primera persona, un estudiant implementa la funcionalitat d'avançar l'observador en la direcció de visió i observa que després d'haver avançat uns quants cops, l'observador mira en sentit contrari al qual avança, la qual cosa és clarament un error.
- a) Això no pot passar, deu tenir un error en la definició de l'escena.
 - b) Deu haver fet un resize i no ha modificat la relació d'aspecte (ra) del window apropiadament i li retalla.
 - c) Deu haver modificat la inclinació (up) de la càmera (tot mantenint la direcció de visió).
 - d) Deu haver modificat OBS per avançar però no VRP.
8. (0.5 punts) Indica quina de les següents inicialitzacions permet obtenir una visualització en planta (projecció en el pla X-Z de l'aplicació) d'una escena de la qual sabem que la seva esfera contenidora està centrada a l'origen i té radi 20.
- a) $VM=I$; $VM=VM \cdot G_x(90)$; $VM=VM \cdot Translacio(0,0,-30)$
 - b) $VM=I$; $VM=VM \cdot Translacio(0,0,-30)$; $VM=VM \cdot G_y(90)$;
 - c) $OBS=(0,50,0)$, $VRP=(0,10,0)$, $up=(0,0,1)$
 - d) $OBS=(0,50,0)$, $VRP=(0,0,0)$, $up=(0,1,0)$
9. (0.5 punts) Definint la viewMatrix utilitzant transformacions geomètriques a partir dels angles d'Euler:
- a) No es pot reproduir l'efecte de modificar el vector up.
 - b) L'òptica obligatòriament ha de ser perspectiva.
 - c) Aquesta matriu no podrà ser la identitat.
 - d) Podem obtenir la mateixa visualització de l'escena que si s'ha obtingut a partir de lookAt.

10. (0.5 punts) En el procés de visualització, l'algorisme de retallat implementat per OpenGL (clipping) per a triangles que es pintaran omplerts de color...
- a) és el mateix algorisme tant si la càmera és perspectiva com si és ortogonal.
 - b) és funció de si el tipus de càmera és perspectiva o és ortogonal.
 - c) és funció de si el volum de visió és un tros de piràmide o un paral·lelepípede.
 - d) és funció de si les coordenades dels vèrtexs a la sortida del vertex shader (gl_Position) estan en coordenades de clipping o no.
11. (0.5 punts) Tenim una funció `pinta_cub` que envia a pintar un cub d'aresta 1 centrat a l'origen de coordenades i una càmera ortogonal que permet visualitzar-ho correctament. Volem que l'usuari, mitjançant una tecla, pugui decidir si vol que el cub es vegi escalat al doble de gran o no. Com ho implementem?
- a) Podem enviar un uniform al FS (amb valors 1 o 2) i, en cas que tingui valor 2, desplaçar els fragments en el viewport per ocupar més espai.
 - b) Sense modificar la posició de la càmera no es pot aconseguir cap escalat.
 - c) Podem enviar un uniform al VS (amb valors 1 o 2) i que es multipliquin les coordenades dels vèrtexs pel uniform.
 - d) Es pot aconseguir aquest efecte reprogramant tant el VS com el FS indistintament.
12. (0.5 punts) Es vol modificar una aplicació per a què cada crida a `paintGL` pinti la mateixa escena en dos viewports diferents. L'escena es pinta mitjançant una crida `pinta_escena()`. Els dos viewports defineixen respectivament el quadrant inferior esquerre i el quadrant superior dret de la finestra gràfica, que és de 800x600 píxels, tots dos tenen la mateixa relació d'aspecte. La posició i orientació de la càmera no es volen modificar. Per a fer això caldrà fer canvis a `paintGL`:
- a) Hem d'afegir una segona crida a `pinta_escena()`, precedint cada crida a `pinta_escena()` de la definició del viewport corresponent.
 - b) Tant sols caldrà modificar la definició del viewport, la `raw` i la `projectMatrix` abans d'una segona crida a `pinta_escena()` que afegirem.
 - c) Tant sols caldrà fer dos crides a `glViewport`, sense necessitat d'afegir res més.
 - d) Afegirem una segona crida a `pinta_escena()`, però redefinirem la `projectMatrix` abans de cadascuna de les dues crides a aquesta funció.
13. (0.5 punts) Un objecte està modelat per un conjunt de triangles:
- a) L'esquema de representació del model pot indicar la topologia (ordre d'unió dels vèrtex) de manera implícita o explícita.
 - b) L'esquema de representació ha de ser sempre una única taula de vèrtexs sense repetició.
 - c) Per intercanviar la informació amb OpenGL utilitzant VBOs (com fem al laboratori) cal que no hi hagi repetició de vèrtexs i que la topologia (ordre d'unió dels vèrtex) sigui explícita.
 - d) L'esquema de representació ha d'indicar sempre la topologia (ordre d'unió dels vèrtex) de manera implícita.

14. (0.5 punts) Els valors per defecte en un formulari...

- a) s'han de posar sempre que hi hagi un percentatge superior al 30% de probabilitat de que sigui el correcte.
- b) són de poca utilitat però els usuaris s'hi han acostumat.
- c) si es posen, cal posar-los per tots els camps.
- d) no necessàriament calen per a tots els camps.

15. (0.5 punts) Tenint en compte que defineix una aplicació per indicar el temps meteorològic, quin tipus de representació es correspon a la icona de la imatge?



- a) Similaritat
- b) Simbòlic
- c) Exemple
- d) Arbitrari