

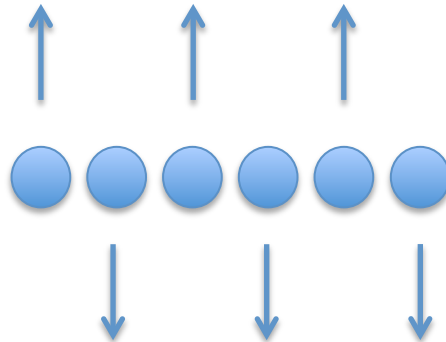
Nom i Cognoms: \_\_\_\_\_ DNI: \_\_\_\_\_ Grup: \_\_\_\_\_

**Normativa**

- Poseu el nom en tots els fulls. Poseu el vostre carnet de la UPC o DNI a la taula.
- No es poden utilitzar ni tenir a la vista ni calculadores ni ordinadors ni dispositius mòbils. Tampoc no es poden utilitzar apunts de cap tipus.
- Responen l'examen en els mateixos fulls.
- Les preguntes de tipus test només tenen una resposta correcta i en cas de contestar-se de forma errònia, tenen una penalització de 0.33.
- Les respostes de totes les preguntes que no siguin de tipus test **han de ser raonades**. Si no estan convenientment justificades, tindran una puntuació de zero.

**Pregunta 1 (1 punt)** Explica en què consisteix la llei del destí comú (*Law of common fate*)

La llei del destí comú és un dels principis de Gestalt que aplicat al disseny d'interfícies diu que els elements que es mouen en la mateixa direcció es perceben com un sol conjunt. Així per exemple, en l'esquema que us vaig mostrar a classe, on hi havia sis cercles dels quals tres es movien cap amunt i tres cap avall:



Els cercles que es mouen cap a dalt són percebuts com un conjunt i els que es mouen cap a baix com un altre grup.

**Pregunta 2 (1 punt)** Segons en Jakob Nielsen, s'ha d'escoltar o no els usuaris en un estudi d'usabilitat? Per què?

En Jakob Nielsen en el seu article "*First rule of usability: Don't listen to users*" defensa que el que s'ha de fer per estudiar la usabilitat és veure com treballen els usuaris, no escoltar el que diuen. Hi ha diverses raons per a no tenir en consideració els comentaris dels usuaris. Entre d'altres, podem mencionar que els usuaris, quan són preguntats davant d'altra gent, poden tendir a contestar el que creuen que espera el que pregunta, o el que pot ser més socialment acceptable. A més, quan se'ls demana per accions o decisions que hagin pres, a banda de tenir una memòria feble, com tothom, pot ser que intentin racionalitzar els seus actes. Tampoc són massa fiables els usuaris quan parlen de característiques de les aplicacions. Podria ser que enfrontats a una interfície que no han provat, mostraran unes preferències que després solen tenir poca correlació amb les predileccions que mostren un cop han provat la interfície. Això no vol dir que no es pugui aconseguir informació valuosa: si es fa un estudi formal i després es presenta un qüestionari de resum, la informació que s'obtingui podrà completar la resta d'informació aconseguida durant l'execució de les tasques. També es pot recollir informació valuosa en els estudis d'usabilitat que utilitzen la tècnica *Think Aloud* perquè en aquest cas és més fàcil entendre les seves accions.

**Pregunta 3 (1 punt)** Quina d'aquestes tasques NO és responsabilitat del *briefe*r en un estudi d'usabilitat.

**Dirigir la confecció de l'informe final.**

**Pregunta 4 (1 punt)** En un entorn de Realitat Virtual...

**La selecció sol ser un problema difícil de solucionar perquè hi ha elements com l'oclusió que en 3D no és trivial.**

**Pregunta 5 (1 punt)** En l'article de George Miller "The Magical Number Seven..."

**L'autor analitza la quantitat d'estímul unidimensionals que les persones poden reconèixer amb la mesura de quantitat d'informació transmesa.**

**Pregunta 6 (1 punt)** La posició i orientació d'una càmera ortogonal està definida mitjançant VRP, OBS i UP. Es visualitza una escena i es vol realitzar un "pan" (translació de l'escena en una direcció perpendicular a l'eix z de l'observador tot simulant un desplaçament de la imatge en la pantalla), a tal efecte cal:

**Modificar VRP i OBS**

**Pregunta 7 (1 punt)** Amb quins valors inicialitzaries les constants empíriques del material d'un objecte que té el següent comportament: els reflexos especulars sempre es veuen del mateix color que la llum del focus i la resta de zones il·luminades pel focus es veuen de color groc si el focus és groc i del mateix color que les zones no il·luminades pel focus quan el focus és de color blau. La llum ambient és blanca.

**$k_d=(1,1,0)$ ,  $k_s=(1,1,1)$ ,  $N=100$  i qualsevol valor de  $k_a$**

Nom i Cognoms: \_\_\_\_\_ DNI: \_\_\_\_\_ Grup: \_\_\_\_\_

**Pregunta 8 (1 punt)** Tenim un rectangle centrat a l'origen de coordenades amb els costats paral·lels als eixos X i Y de l'aplicació i de mides 4cm d'ample i 3cm d'alçada. Tenim una càmera inicialitzada amb OBS=(0,0,5), VRP=(0,0,0), VUV=(0,1,0). El *viewport* és de 400 píxels d'amplada i 600 d'alçada. Indiqueu els paràmetres amb què cal inicialitzar *gluPerspective()* de manera que encara que fem girar el rectangle respecte l'eix Z de l'aplicació sempre es visualitzi sense deformacions en el *viewport* i sense ser retallat.

Nota: la càmera estarà fixa, és a dir, encara que girem el rectangle NO modifiquem la càmera.

Com el rectangle es troba centrat en l'origen de coordenades i només pot girar respecte a l'eix Z, la seva distància a l'observador serà sempre de 5. Per tant:

-  $5 \geq zNear \geq 0$  per tal de poder veure el rectangle; per exemple:  $zNear=5$

-  $zFar \geq 5$  per tal de que el rectangle estigui dins del camp de visió; per exemple:  $zFar=6$

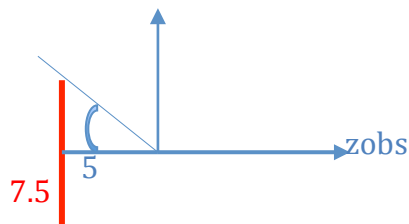
La relació d'aspecte de la càmera (window) ha de ser la del viewport per a què no hi hagi deformacions; per tant:  $ra=400/600=2/3$ .

Com el rectangle pot girar nomé a l'entorn de eix Z, cal assegurar un angle d'obertura de la càmera que englobi el seu cercle contenidor per a que no es retalli en cap de les seves ubicacions. El radi d'aquest cercle serà:  $\sqrt{4^2 + 3^2}/2 = 2.5$ . Per tant, el window mínim serà de 5x5 ( $ra^*=1$ ). Com la  $ra < 1$ , caldrà incrementar l'alçada per a què no hagi deformacions i tenir la ra requerida:

$alçada = amplada/ra = 5/(2/3) = 7.5$ .

Cal un angle mínim que garanteixi un window d'alçada 7.5; d'acord amb les dades d'ubicació de SCO:

$$7.5/2=5 \operatorname{tg}(FOV/2) \Rightarrow FOV = 2 * \operatorname{arctg}(7.5/10) = 2 * 36 = 72^\circ$$



**Pregunta 9 (1 punt)** S'aproxima un cilindre per un prisma amb bases poligonals de 20 costats. S'inicialitza una càmera en una posició arbitrària que permet veure el cilindre optimitzant la seva ocupació en el viewport. Es visualitza utilitzant OpenGL amb z-buffer, il·luminació i *shading* de *Gouraud* (*Smooth*). Quines diferències qualitatives observaries en la imatge resultant si s'envia a pintar la geometria amb normal per vèrtex o normal per cara?

Si utilitzem normal per cara, tots els seus vèrtexs tindran la mateixa normal (la de la cara). En calcular la il·luminació en els vèrtexs d'una cara, en funció de la direcció de la llum incident en ells i de la posició de l'observador, els seus colors poden ser diferents i en la cara podem trobar una gradació de colors. Tanmateix, en cares veïnes que comparteixen vèrtexs, el color d'un vèrtex comú serà diferent per a cada cara perquè tenen diferent normal. Per tant, al llarg d'una aresta comú, quedaran colors diferents i en la visualització es veurà un salt de color entre cares. Es visualitzarà com un prisma (es notaran les arestes).

Si utilitzem normal per vèrtex, entenem que la normal en cada vèrtex és una ponderació de les normals de les cares que coincideixen en ell. Per tant, en cada vèrtex d'una mateixa cara tindrem normal diferents i dins d'una cara tindrem un gradació de colors. A més a més, com cares veïnes comparteixen vèrtexs, en aquests tindrem el mateix color per les cares que incideixen en ell. Per tant, al llarg d'una aresta comú quedaran colors similars en les dues cares que la comparteixen, i no observarem el salt de color entre cares (es "suavitzen" les arestes i no es veuen), obtenint una visualització més semblant a un cilindre. Tanmateix, a la silueta de l'objecte, s'observaran les arestes perquè al pintar-se una sola cara, veurem el salt de colors entre la cara i el color de fons de la imatge. Ens adonarem que el cilindre s'ha aproximat per cares planes.

Nom i Cognoms: \_\_\_\_\_ DNI: \_\_\_\_\_ Grup: \_\_\_\_\_

**Pregunta 10 (1 punt)** El següent codi permet pintar un cub de costat 6 centrat en el punt (0,3,0) amb cares paral·leles als plans coordenats. El volem il·luminar amb dues llums puntuals. La llum 0 s'ha de moure amb la càmera i ha d'estar situada 10 unitats per sobre de l'observador. La llum 1 estarà ubicada en la recta que passa pel centre del cub en la direcció dels Z positius de l'aplicació i a distància 6 del centre del cub.

1. `setProjection();` //inicialitza la `gluPerspective` correctament per a veure el cub
2. `glMatrixMode(GL_MODELVIEW);`
3. `glLoadIdentity();`
4. `glTranslatef(0,0,-6);`
5. `glRotatef(-90,0,1,0);`
6. `glTranslatef(0,-3,0);`
7. `Pinta_Cub();`

Indica raonadament les crides a *glLight* necessàries per a posicionar ambdues llums com s'indica i de forma que en cada visualització la llum 1 apunti a una cara lateral diferent del cub (giri al voltant de l'eix Y de l'aplicació). Indica clarament el lloc del codi en que les posaries. Observació: els colors de les llums ja estan inicialitzats, només has de fer la crida per a indicar la posició.

La llum 0 s'ha de moure amb la càmera, per tant, cal donar la seva posició directament respecte del sistema de coordenades de l'observador (0,10,0). A tal efecte, declararem la seva posició abans de declarar la càmera quan tenim en la ModelView la identitat (entre les instruccions 3 i 4 del codi):

```
GLfloat pos0[4]={0.0, 10.0, 0.0, 1.0}; //focus puntual a (0,10,0) d'Obs.  
glLightfv(GL_LIGHT0, GL_POSITION, pos0);
```

La llum 1 ha d'ubicar-se en una posició concreta de l'escena i ha de ser fixa respecte aquesta, per tant, és una llum d'escena i cal declarar la seva posició un cop tenim la càmera posicionada, entre les instruccions 6 i 7. Inicialment estaria en la posició (0,3,6).

Després de cada nova visualització cal anar girant aquesta llum respecte a l'eix Y de l'escena. Una manera de fixar cada nova posició és rotar un angle de 90° respecte a l'eix Y la posició inicial aprofitant la ModelView. Ho podem aconseguir amb el següent codi:

```
GLfloat pos1[4]={0.0, 3.0, 6.0, 1.0}; //focus puntual a (0,3,-6) en SCA.  
glPushMatrix();  
glRotatef(90*(i%4),0,1,0);  
glLightfv(GL_LIGHT1, GL_POSITION, pos1);  
glPopMatrix();  
i=i+1;
```

Observació: les declaracions de les posicions inicials `pos0` i `pos1` poden anar a l'inici del codi. La variable "i" estarà inicialitzada a valor 0.