

Normativa

- Poseu el nom en tots els fulls. Poseu el vostre carnet de la UPC o DNI a la taula.
- No es poden tenir a la vista ni calculadores ni ordinadors ni dispositius mòbils.
- Responen l'examen en els mateixos fulls.
- Les preguntes tipus test valen 0.5 punts i, en cas de resposta incorrecta, resten 0.15.
- Les respostes de totes les preguntes que **no són tipus test han de ser raonades**.

Preguntes 1 a 5:

Pregunta (0.5 punts) Quin dels següents càrrecs NO representa a una de les persones que participen en la realització dels estudis d'usabilitat (els noms estan en anglès com en els apunts per evitar ambigüitats):

Solució: Final reporter.

Pregunta (0.5 punts) Les icones de la interfície que es dissenya:

Solució: En general, no s'han de redissenyar si serveixen per il·lustrar tasques comunes que l'usuari ja coneix.

Pregunta (0.5 punts) Les entrevistes posteriors als tests d'usabilitat:

Solució: Cap de les anteriors.

Pregunta (0.5 punts) El següent tros de codi defineix la posició i orientació d'una càmera; quina seria la definició de la mateixa càmera utilitzant *gluLookAt(...)*?

```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
glTranslate(0, 0, -2);  
glRotate(90, 0, 1, 0);  
glRotate(90, 0, 0, 1);
```

Solució: gluLookAt(0,2,0,0,1,0,1,0,0)

Pregunta (0.5 punts) Els dispositius mòbils d'avui en dia:

Solució: Pateixen del problema anomenat del "dit gros", que vol dir que els dits de l'usuari, quan són grossos redueixen la mida efectiva de la pantalla, encara que la resolució sigui molt alta.

Pregunta 6 (1 punt) Descriu el *chunking* i explica en què es fonamenta.

La paraula *chunk* es refereix a una unitat d'informació en memòria de curt termini. *Chunking* és la tècnica que intenta distribuir la informació de forma que s'acomodi als límits que els humans tenim per processar trossos d'informació. En el seu famós article, George Miller estudia les capacitats del cervell humà per a recordar coses. La conclusió, sintèticament, és que ens costa menys recordar trossos petits que llargs, per exemple, molta gent pot recordar una llista de 5 paraules durant 30 segons, però poca gent és capaç de recordar una llista de 10 paraules durant 30 segons.

Com que aquesta tècnica es refereix a elements que cal recordar, com ítems de menús, nombres de telèfon, etc., el que fa és proposar la divisió d'aquests elements en grups de 5, aproximadament.

Altres tasques, com la cerca en un text, no requereixen recordar el text, i per aquesta raó, elements com pàgines d'un diccionari, no s'han de tractar d'aquesta forma.

Pregunta 7 (1 punt) Descriu el concepte de “Non-perceived affordances” i indica si ho has trobat en alguna de les pràctiques a avaluar.

Aquest concepte es refereix a la carència d'elements en una interfície que indiquin el que es pot fer amb aquella interfície. El concepte és important perquè en una interfície, tots els píxels són susceptibles de ser clicats, tot i que en molts casos, aquest clic no origini cap acció. En conseqüència, el dissenyador ha d'evitar que l'usuari es dediqui a clicar aquí i allà a veure si passa quelcom.

Un exemple clar és la pràctica de la llista de *ToDo's* que té botons que estan representats com etiquetes, en comptes d'utilitzar un *widget* clàssic de botó. Els botons, així com els *checkboxes*, són d'un ús evident a partir del seu aspecte (en particular, també hi ha una etiqueta que fa la feina de *checkbox* en aquesta pràctica), mentre que les etiquetes no necessàriament indueixen a l'usuari a fer-hi clic.

Pregunta 8 (1 punt) Un estudiant afirma en un examen que tant el disseny de menús a la part de dalt de la pantalla com la localització dels menús a la part de dalt de cada finestra tenen una justificació en la llei de Fitts. Digues si l'estudiant té raó o s'equivoca i justifica la teva resposta de forma acurada.

L'estudiant té raó. La llei de Fitts expressa el temps que es triga en fer clic a un element que es pugui seleccionar com una fórmula ($MT = a + b ID$), on $ID = \log_2 \left(\frac{2A}{W} + 1 \right)$ que depèn de la distància del desplaçament del punter, o amplada del moviment (A) i la magnitud de l'objectiu en la direcció del desplaçament (W). MT creix si el moviment a realitzar és més ampli, i decreix si la magnitud de l'objectiu creix, perquè es requereix menys precisió.

En conseqüència, ambdues distribucions dels menús estan fonamentades en la llei de Fitts, però de forma diferent. La localització dels menús a dalt de la pantalla facilita la seva selecció perquè la magnitud de l'objectiu (W), en direcció vertical és virtualment infinita, ja que el punter, un cop arribat a la part de dalt, deixa de desplaçar-se si continuem movent el ratolí cap amunt. El resultat és que la dimensió efectiva de l'element a seleccionar és molt gran i, en conseqüència, hem de trigar menys a seleccionar-lo comparat amb un menú amb títols de menú de les mateixes dimensions però que no estigui enganxat a la part superior de la pantalla. Ara bé, en pantalles molt grans, la distància a recórrer per arribar a la part de dalt pot arribar a ser molt gran.

En el segon cas, situar els menús a les diferents finestres redueix la distància a recórrer per part del ratolí (A), ja que es suposa que quan volem escollir un menú, el menú que volem visitar té relació amb la tasca que estem treballant, i per això, molt probablement, ens dirigirem al menú que estigui en la part superior de la finestra de treball. En conseqüència, com que el menú segurament serà proper, el valor de MT també ha de baixar.

Pregunta 9 (1 punt) Donada una càmera en primera persona perspectiva (com la ubicada en la pràctica en el cap del dofí) i un *viewport* quadrat, quins dels seus paràmetres caldrà modificar en els següents casos:

- S'avança cap al davant de la càmera (en la direcció de visió).
- Es modifica la relació d'aspecte de la pantalla (és fa un *resize*).
- Es vol realitzar un *zoom*.
- S'inclina la càmera, mantenint la mateixa direcció de visió.

Raona la resposta en base a la mateixa definició de la càmera en tots els casos, usant *gluPerspective()* i bé *gluLookAt()* o bé transformacions geomètriques.

- Només cal modificar, en *gluLookAt()*, *OBS* i *VRP* en la direcció de visió actual $v = \text{norma}(\text{VRP} - \text{OBS})$ en funció del desplaçament efectuat λ ; $\text{OBS} = \text{OBS} + \lambda v$; $\text{VRP} = \text{VRP} + \lambda v$.
- Només cal modificar en *gluPerspective()* la *ra* de la càmera amb el nou valor de la *ra* *viewport* de la pantalla per a què no hi hagi deformacions.
- Només cal modificar l'angle d'obertura de la càmera *FOV* en *gluPerspective()*, incrementant-lo o disminuint-lo segons *zoom-out* o *zoom-in*.
- Donat que només cal modificar *yobs* (direcció vertical de la càmera) tot deixant la mateixa direcció de visió, només caldrà modificar el vector *up* en *gluLookAt()*.

Pregunta 10 (1.5 punts) Un terra molt polit (brillant) de color blau es modela amb un polígon de coordenades (0,0,10), (10,0,10) (10,0,0) i (0,0,0) i normal (0,1,0). Existeix un focus de llum blanc d'escena a la posició (5,5,5). Es visualitza utilitzant un codi OpenGL que utilitza "*smooth shading*", una càmera ubicada en (10,5,10) que mira cap al (5,0,5) i que permet veure tot el polígon, indica i justifica:

- Les constants empíriques del material del terra.
- La distribució de colors que veurem en el terra

Si ara el terra es pinta amb 100 quadrats de 1x1 que cobreixen la mateixa àrea que el polígon original:

- La distribució de colors que veurem en el terra

Donat que es tracta de un material molt polit, tindrà força reflexió especular, per tant: $k_s = (1,1,1)$ i $N=100$; com és de color blau la reflexió difusa serà blava, per tant, per exemple: $k_d = (0,0,0.5)$ i $k_a = (0,0,0.3)$.

OpenGL calcula el color en els vèrtexs. Donat que tots els vèrtexs tenen la mateixa normal i que el focus està centrat respecte del polígon, l'angle θ entre la direcció d'incidència de la llum en ells i la seva normal és per tots el mateix; per tant, els 4 vèrtexs emeten la mateixa reflexió difusa blava: $(1,1,1) * (0,0,0.5) * \cos(\theta) = (0,0,0.5) * \cos(\theta)$ –suposem que no hi ha llum ambient-. Encara que el material sigui polit, degut a la posició de l'observador, aquest no pot observar la reflexió especular en cap vèrtex (l'angle entre la direcció de reflexió de la llum en ells i la direcció de visió dels vèrtex és gran). Com a conclusió, encara que fem *smooth shading*, com els 4 vèrtexs tindran el mateix color, el polígon es pintarà d'un blau constant.

Ara hi ha molts més vèrtexs on es calcularà el color, tots amb la mateixa normal; però amb angles θ diferents. Pels quadrats ubicats sota de la llum aquest angle serà més petit i, per tant, la seva reflexió difusa serà un blau més brillant que pels vèrtexs dels quadrats que es troben cap a la frontera del terra. Això comporta, que el terra es veurà amb un degradat de blaus, més brillant al centre i enfosquit cap a la frontera. A més, l'observador podrà veure una taca especular blanca en la zona al voltant del punt (7.5, 0, 7.5) ja que la direcció de visió dels vèrtexs dels quadrats d'aquesta zona estarà dins del seu conus de reflexió especular.

Pregunta 11 (0.5 punt) Una escena amb dues esferes A i B (representades cadascuna per un poliedre convex) es mira des d'una posició on l'esfera B queda davant de la A i la tapa parcialment. Usant per a l'eliminació de parts amagades NOMÉS l'algorisme de *back-face culling*, ens trobem que la visualització obtinguda no és correcta perquè el tros de l'esfera A que ha de ser tapat per l'esfera B es veu en la imatge.

- Per què no està funcionant correctament l'eliminació de parts amagades?
- Com ho solucionaries?

Perquè el *Back-face culling* només elimina aquelles cares dels poliedres en què l'observador es troba en el semi-espai interior del seu pla suport. Per cada cara considera només el seu pla i la posició de l'observador. Per tant, per un poliedre convex elimina totes les seves cares no visibles si suposem que no hi ha altre objecte en l'escena. Si hi ha més d'un, no té en compte les cares d'un (B) que tapen l'altre (A); per tant, si es pinta B i després A, cares de A que haurien de quedar tapades per B quedaran visibles.

Es pot solucionar aplicant, a més a més, el z-buffer.

Pregunta 12 (1.5 punts) El codi següent pinta una escena formada per un cub de material blanc mate (no cal considerar la reflexió especular) d'aresta 20 centrat a l'origen de coordenades. Assumint el següent: el *depth-test* i la il·luminació estan habilitats, el focus de llum 1 és de color blau, el focus de llum 2 de color groc, el *viewport* quadrat i *pintacub()* conté totes les crides OpenGL requerides per a pintar el cub.

```
float posllum1[4] = {0, 0, 10, 1};
float posllum2[4] = {0, 30, 0, 1};
glMatrixMode (GL_PROJECTION);
glLoadIdentity();
gluPerspective (90, 1, 5, 40);
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
glLightfv (GL_LIGHT1, GL_POSITION, posllum1);
glPushMatrix ();
* gluLookAt(...);
glLightfv (GL_LIGHT2, GL_POSITION, posllum2);
pintacub();
glPopMatrix();
```

Indiqueu com es veuria en pantalla colorat el cub en les següents casos:

- si en * posem `gluLookAt(20,0,0,0,0,0,0,1,0)`
- si en * `gluLookAt(0,20,0,0,0,0,0,0,1)`

Per la seva ubicació en el codi, el focus 1 és de càmera i sempre estarà ubicat en la posició (0,0,10) respecte el sistema de coordenades de l'observador, per tant, la seva ubicació respecte el cub canviarà al moure la càmera. En canvi, el focus de llum 2 és d'escena i està en la posició (0,30,0); la seva posició és fixa respecte al cub.

En totes dues ubicacions, el cub està dins del volum de visió de la càmera i l'observador veurà una sola cara del cub.

Quan tenim `gluLookAt(20,0,0,0,0,0,0,1,0)`, l'observador està ubicat sobre l'eix X_A mirant al (0,0,0) i, per tant, el focus 1 està a la posició (30,0,0) respecte l'escena. La cara visible per l'observador és la ubicada en el pla $X=10$; la normal en els seus vèrtexs és (1,0,0). Donada la posició del focus 2, l'angle θ entre la direcció d'incidència de la llum als vèrtexs i la seva normal és més gran de 90° , per tant, no els il·lumina. Els vèrtexs només reben la llum del focus 1 (blau) tots amb el mateix angle θ ; per tant, la cara es veurà de color constant blau (donat que el cub és blanc).

Quan tenim `gluLookAt(0,20,0,0,0,0,0,0,1)` l'observador està ubicat sobre l'eix Y_A mirant al (0,0,0) i, per tant, el focus 1 està a la posició (0,30,0) respecte l'escena. Noteu que és la

mateixa posició que el focus 2. La cara visible per l'observador és la ubicada en el pla $Y=10$; la normal en els seus vèrtexs és $(0,1,0)$. Donada la posició dels focus, l'angle θ entre la direcció d'incidència de la llum als vèrtexs i la seva normal és menor de 90° , per tant, els dos focus els il·luminen per igual. Els vèrtexs rebem llum blanca (blau+groc) tots amb el mateix angle θ ; per tant, la cara es veurà de color constant blanc (donat que el cub és blanc).