Examen parcial de VIG. Curs 2005-2006 (2Q) Temps: 1h 45m

Noms i Cognoms: DNI:

1. (2.5 punts) Es disposa d'una acció *pinta_esfera (c, r)* que, a partir de les coordenades del *centre* i del valor del *radi* d'una esfera, genera i pinta utilitzant OpenGL els triangles que la modelen. Es desitja visualitzar un sistema solar molt simple: el sol, els planetes i els satèl·lits es consideren esferes, i les òrbites dels planetes i dels satèl·lits es consideren circulars essent els eixos de gir paral·lels a l'eix *y* del sistema de coordenades de l'escena i passant pel centre del sol (en el cas dels planetes) o dels planetes (en el cas dels seus satèl·lits).
   Es demana:

a) Dissenya una estructura de dades que permeti emmagatzemar la informació requerida per a programar un tros de codi que, utilitzant *pinta_esfera*, permeti obtenir imatges del sistema solar en diferents instants de temps.

b) Suposant que teniu definida una camera, dissenyeu el tros de codi que recorre l'estructura de dades dissenyada i utilitzant *pinta_esfera(c,r)* i comandes OpenGL genera la imatge del sistema solar en un instant de temps determinat.

2. (1 punt) Donat un cub de longitud d'aresta igual a 2, centre en l'origen de coordenades i costats paral·lels als eixos de coordenades, indiqueu les coordenades dels seus vèrtexs (-1,-1,-1) i (1,1,1) respecte del sistema de coordenades de l'observador, en coordenades normalitzades i en coordenades de dispositiu (en aquest cas només la x i la y). Suposeu que s'ha definit un viewport del (0,0) al (100,100) i una càmera axomomètrica amb els següents paràmetres: Obs=(2,0,0), VPR(0,0,0), VUP=(0,1,0), window del (-1,-1) al (1,1), zN=1 i zF=3.

3. (1.5 punt) Considera aquest fragment de codi que defineix una càmera perspectiva:

```
1. glMatrixMode(GL_PROJECTION);
2. glLoadIdentity();
3. gluPerspective(60, 1, znear, zfar);
4. glMatrixMode(GL_MODELVIEW);
5. glLoadIdentity();
6. glTranslatef(0, 0, -20);
7. glRotatef(90, 0, 1, 0);
8. glTranslatef(-10, -10, -10);
9. drawModel();
```

Indica i justifica els valors dels paràmetres Obs, VRP i up que hauries d'utilitzar per a definir la mateixa càmera amb gluLookAt(...);

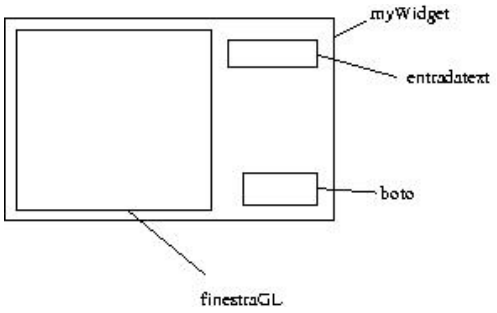
4. (1 punt) Considera aquest fragment de codi:

```
1. glViewport(0, 0, w, h);
2. glMatrixMode(GL_PROJECTION);
3. glLoadIdentity();
4. gluPerspective(60, ...);
5. glMatrixMode(GL_MODELVIEW);
6. glLoadIdentity();
7. glTranslatef(0, 0, -10);
8. drawSphere(5);
```

on la funció *drawSphere(5)* envia a OpenGL els vèrtexs d'una esfera de radi 5 centrada en l'origen. El viewport és quadrat (w = h). Quins valors donaries a la resta de paràmetres de gluPerspective per tal d'obtenir com a resultat una el·lipse amb el doble d'alçada que d'amplada?

5. (1 punt) Volem aconseguir una interfície en Qt com la de la figura.
   a) Indica quins són els components de Qt necessaris, incloent els spacers i layouts que calguin.

   b) Què caldria fer per a que cada cop que s'entri la tecla 'X' en el component entradatext s'executi el mètode *do_it()* -sense cap paràmetre- del component finestraGL. Explica-ho i implementa el que calgui (no cal que la sintaxi sigui exacta, però sí que s'entengui).



6. (2 punts) Tenim una escena amb un conjunt de cotxes que es van movent per un terreny horitzontal (y=0). Disposem d'una rutina que ens retorna les coordenades dels punts centrals dels dos cotxes que estan a mínima distància un de l'altre. Volem definir una càmera que permeti veure aquests dos cotxes a la vista (*viewport*). Especifiqueu la càmera, si fem la hipòtesi de que la recta OBS-VRP està en el pla y=0. Podeu considerar que el radi de l'esfera englobant de qualsevol cotxe, centrada al seu punt central, és *Rc*.

7. (1 punt) En què consisteix el procés de retallat de primitives? Es pot utilitzar directament l'algorisme de retallat de punts per a retallar segments?