Exercici 35: Dos codis per a una vista en planta:

Codi 1

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
glOrtho (-100 ,100 ,-100,100 ,10,150);
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
gluLookAt (0,80,0,0,50,0,1,0,0);
```

Preguntes:

a) *Defineixen els 2 vista en planta?*

b) *Defineixen la mateixa càmera?*

c) *Es poden optimitzar les TGs del codi 2?*

Codi 2

```
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
glOrtho (-100 ,100 ,-100,100 ,10,150);
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
glTranslatef (0, 0, -80);
glRotatef (90 , 0, 0, 1);
glRotatef (90 , 1, 0, 0);
glRotatef (-90, 0, 1, 0);
```

Exercici 62: Una esfera de radi 1 es visualitza en un viewport quadrat de 400 per 400, amb una càmera posicionada correctament amb angles d'Euler, i on el mètode per a definir la projecció de la càmera utilitza la següent crida:

```
gluPerspective(60.0, 1.0, 1.0, 10.0);
```

L'usuari ha redimensionat la finestra a 500 d'amplada per 400 d'alçada. Digues què cal canviar de la càmera per tal que es vegi l'esfera correctament (sense retallar-la ni deformar-la).

a) Incrementar l'angle d'obertura vertical (FOV) i la relació d'aspecte del window.
b) Augmentar la relació d'aspecte del window i la distància al ZNear.
c) Només augmentar la relació d'aspecte del window.
d) Només canviar l'angle d'obertura vertical (FOV).

Exercici 58: Indica quina de les inicialitzacions de l'òptica perspectiva és més apropiada per a una càmera que porta un observador que camina per una escena fent fotos. Esfera englobant d'escena té radi R, d és la distància entre OBS i VRP. Observació: ra_v és la relació d'aspecte del *viewport*

a) FOV = 60º, ra = ra_v, zNear = 0.1, zFar = 20
b) FOV = 60º, ra = ra_v, zNear = R, zFar = 3R;
   essent R el radi de l'esfera contenidora de l'escena.
c) FOV = 2*(arcsin(R/d)*180/PI), ra = ra_v, zNear = R, zFar = 3R;
   essent R el radi de l'esfera contenidora de l'escena i d la distància d'OBS a VRP.
d) FOV = 2*(arcsin(R/d)*180/PI), ra = ra_v, zNear = 0, zFar = 20;
   essent R el radi de l'esfera contenidora de l'escena i d la distància d'OBS a VRP

Exercici 71: Disposem d'una càmera axonomètrica amb els següents paràmetres:

OBS=(0.,0.,0.), VRP=(-1.,0.,0.), up=(0.,1.,0.),

window de (-5,-5) a (5,5), zn=5, zf=10.

Indiqueu quin altre conjunt de paràmetres de càmera defineix exactament el mateix volum de visió (és a dir, garanteix generar exactament la mateixa imatge de l'escena):

a) OBS= (1,0,0), VRP= (0,0,0), up=(0,2,0), zn= 6, zf=11
b) OBS= (0,1,0), VRP=(0,0,0), up= (0,1,0), zn=5, zf=10
c) OBS= (0,0,0), VRP=(-2,0,0), up=(0,1,0), zn=6, zf=11
d) OBS= (-1,0,0), VRP=(0,0,0), up=(0,1,0), zn=-1, zf=9

Exercici 67: Disposem d'una càmera ortogonal amb els següents paràmetres:

OBS=(0.,0.,0.), VRP=(-1.,0.,0.), up=(0.,1.,0.), window de (-5,5) a (5,5), ra=1, zn=5, zf=10.

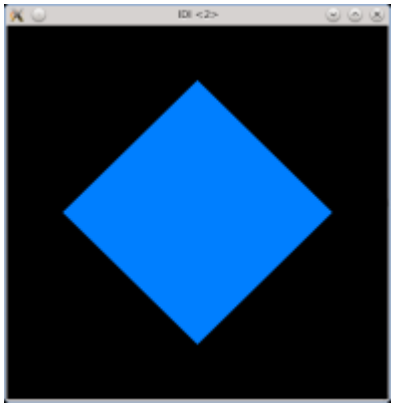
Indiqueu quin conjunt de paràmetres d'una càmera perspectiva defineix un volum de visió que conté l'anterior (és a dir, garanteix que es veurà, coma mínim, el mateix que amb la càmera axonomètrica):

a) FOV= 90, ra=1, zn= 5, zf=10
b) FOV= 60, ra=1, zn=5, zf=10
c) FOV= 60, ra= 2, zn=6, zf=11
d) FOV= 90, ra= 0.5, zn=5, zf=10

Exercici 82: Pintem una escena amb el següent codi:

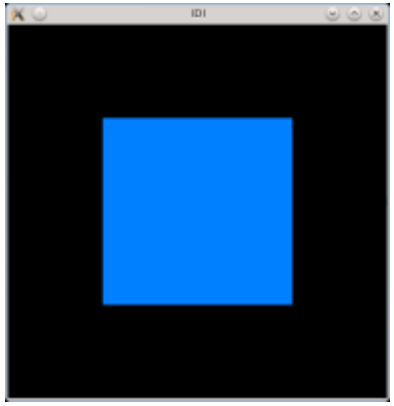
```
glClearColor (0.0, 0.0, 0.0, 1.0);
glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
glOrtho (-2.0, 2.0, -2.0, 2.0, -10., 15.);
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
glPushMatrix ();
glTranslatef (0, -1, -10.);
glRotatef (-90.0, 1.0, 0.0, 0.0);
glColor3f (0.0, 0.5, 1.0);
glutSolidCone (1.0, 1.0, 20, 20);
glPopMatrix ();
glPushMatrix ();
glTranslatef (0, 1, -10.);
glRotatef (90.0, 1.0, 0.0, 0.0);
glutSolidCone (1.0, 1.0, 20, 20);
glPopMatrix ();
```

a)



b)

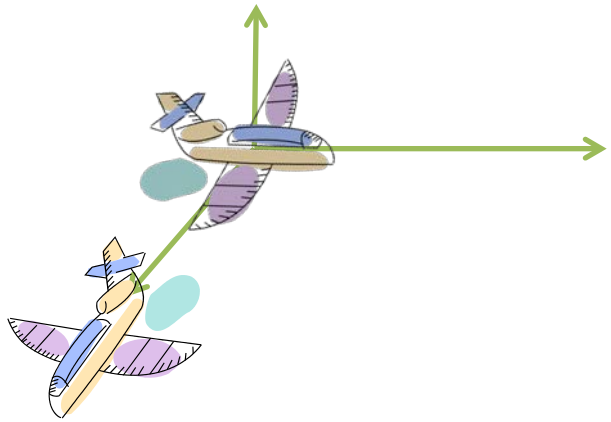
c)



d)

Quina imatge es veurà? (glutSolidCone orientat en Z+)

Exercici 70: Una escena està formada per dos avions. Un avió tindrà el centre de la seva capsa mínima contenidora en el (0,0,0) i estarà orientat cap l'eix X+; l'altre avió tindrà en centre de la seva capsa mínima contenidora en (0,0,120) i orientat cap l'eix Z+. La llargada dels avions és 100 i de punta a punta de les ales 100.

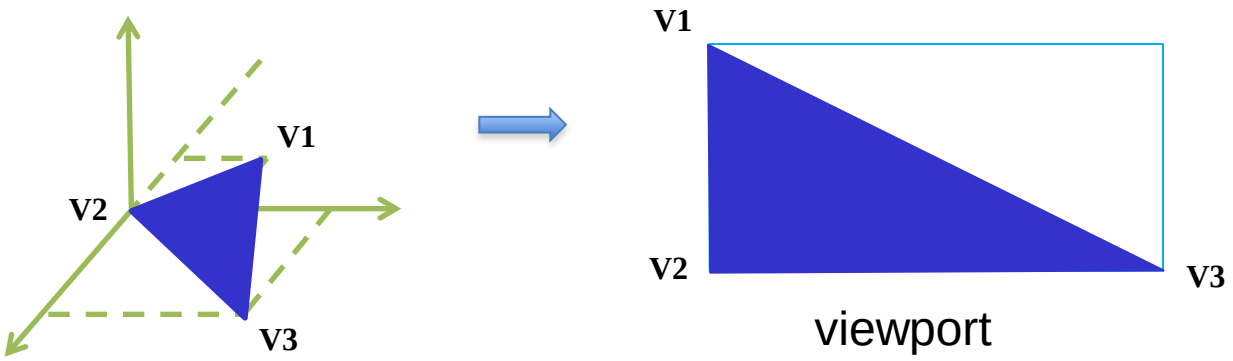


Indica els paràmetres d'una càmera ortogonal (posició + orientació amb transformacions geomètriques, i òptica) i codi OpenGL que permeti veure només el primer dels dos avions en una vista que permeti veure tant el seu cilindre central com les seves "ales". Dibuixa la imatge resultant i justifica l'elecció de tots els paràmetres. El viewport és quadrat.

Exercici 64: Quan s'inicialitza la càmera, en quin ordre cal indicar les transformacions de càmera i el viewport a OpenGL?

a) No importa l'ordre en què s'indiquen.
b) Transformació de posició + orientació, transformació de projecció, *viewport*.
c) La transformació de projecció, transformació de posició + orientació, *viewport*.
d) *Viewport*, transformació de projecció, transformació de posició + orientació.

Exercici 45: Tenim un triangle rectangle amb els vèrtexs V1=(2,0,-2), V2=(0,0,0) i V3=(4,0,4). Escriu el codi OpenGL que defineix una càmera perspectiva de manera que els vèrtex en Sistema de Coordenades de Dispositiu (SCD) en un viewport de 800x400 píxels siguin V1scd=(0,400), V2scd=(0,0) i V3scd=(800,0). Defineix els paràmetres de posició i orientació mitjançant gluLookAt i transformacions geomètriques.



Cal definir la posició i orientació de la càmera i l'òptica