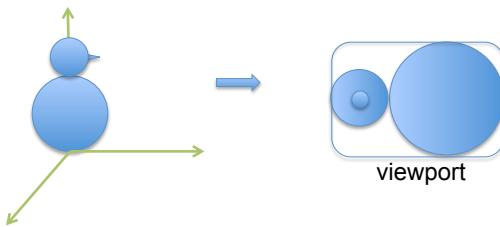


Exemple 1: Donada l'escena de la figura, formada per: una esfera de radi 10 i centre (0,10,0), altre esfera de radi 5 i centre (0,25,0), i un con de base centrada en (2.5, 25, 0), r=2 i llargada 5 orientat segons l'eix X; iniciar tots els paràmetres d'una càmera ortogonal que permeti obtenir la imatge que s'indica en un viewport de 600x400.

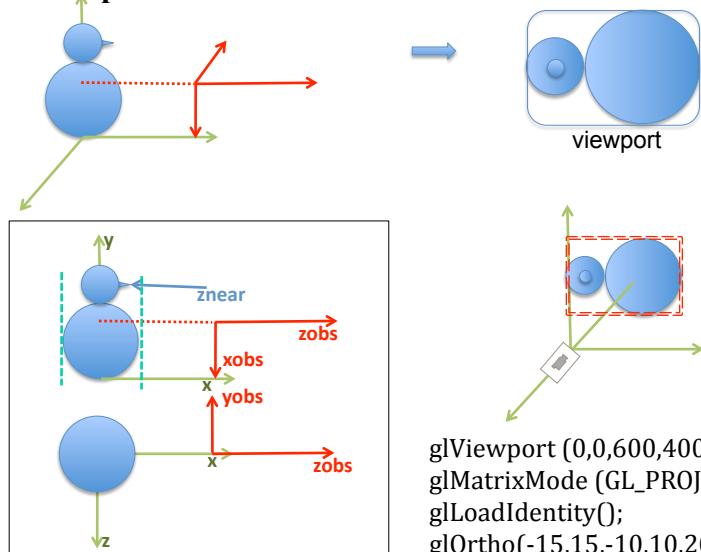


```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0.,0.,-30.);
glRotatef(90.,0.,0.,1.);
glRotatef(-90.,0.,1.,0.);
glTranslatef(0.,-15.,0.);
Pinta_ninot_de_neu; //
```

IDI 2013-2014 2Q

20

Exemple 1: Optica axonomètrica



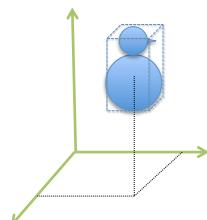
```
glViewport(0,0,600,400);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-15,15,-10,10,20,40);
```

IDI 2013-2014 2Q

21

Exercici d'inicialització i moure càmera

- Veure escena sempre sense retallar i sense deformació (en pas a viewport)
- La imatge inicial volem que estigui centrada i ocupant raonablement el viewport i mostri escena vista des d'una posició arbitrària.
- Càmera perspectiva.
- Permetre modificació interactiva de punt de vista.

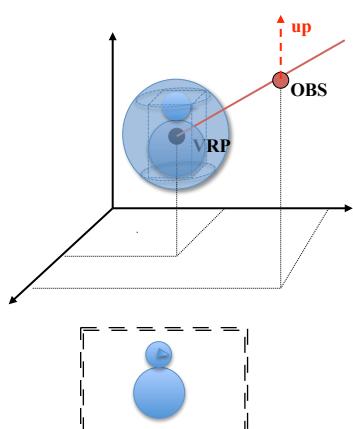


Imaginem aquesta escena

IDI 2013-2014 2Q

22

Exercici d'inicialització posicionament amb OBS, VRP, up

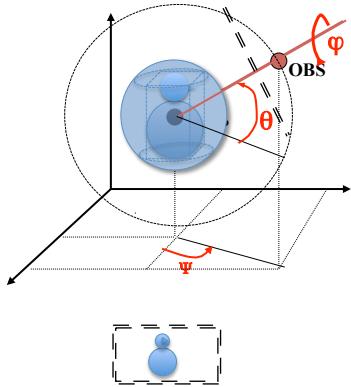


- Centrat => VRP=CentreEscena
 - Per assegurar que escena es veu sense retallar des d'una posició arbitrària CAL que OBS sempre forà capsa mínima contenidora en una posició qualsevol; per assegurar-ho CAL que OBS forà de l'esfera englobant de la capsà => distància "d" de l'OBS a VRP superior a R esfera.
- CapsaMinCont=(xmin,ymin,zmin,xmax,ymax,zmax)
 CentreEscena=Centre(CapsaMinCont)=> ((xmax+xmin)2, ...)
 $R=\text{dist}((xmin,ymin,zmin),(xmax,ymax,zmax))/2$
 $d=R$; per exemple $d=2R$
 $OBS=VRP+d*v$; v normalitzat en qualsevol direcció;
 per exemple $v=(1,1,1)/\|(1,1,1)\|$
- up qualsevol que no sigui paral·lel a v; si volem ninot vertical (eix Y és vegi vertical) up=(0,1,0)

IDI 2013-2014 2Q

23

Exercici d'inicialització càmera: Posicionament amb angles Euler (TG)



- Imaginemem movem la càmera (OBS) sobre una esfera centrada en VRP de radi d.
- VRP i d calculats com s'ha vist en exercici anterior.
- Podem definir la posició amb angles Euler: ψ i θ .
- per exemple: $\psi = 45^\circ$ i $\theta = 45^\circ$ (o altres valors)
- gir càmera sobre si mateixa: φ
- per exemple: 0° per veure eix Y vertical.

- Podem indicar càmera a OpenGL amb TG a objecte (més directe) o calculant VRP, OBS i up (veure transpa següent). Recordeu que transformacions "en codi" són als objectes (repasseu exercicis anteriors)

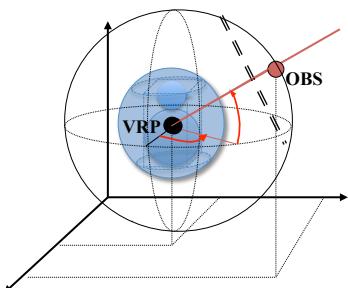
```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0.,0.,-d)
glRotatef(0,1,0,0)
glRotatef(-psi,0,1,0.)
glTranslatef(-VRP.x,-VRP.y,-VRP.z))
```

Ull amb signes:

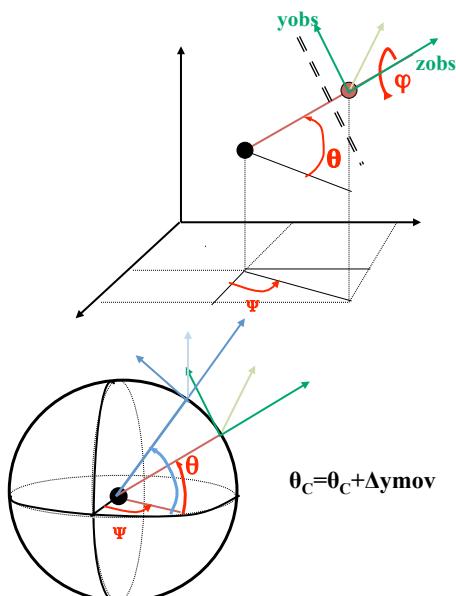
- Si s'ha calculat ψ positiu quan càmera gira cap a la dreta, serà un gir anti-horari respecte eix Y de la càmera, per tant, matemàticament positiu; com girem els objectes en sentit contrari, cal posar $-\psi$ en el codi.
- Si s'ha calculat θ positiu quan pugem la càmera, serà un gir horari; per tant, matemàticament un gir negatiu; com objecte girarà en sentit contrari (anti-horari), ja és correcte deixar signe positiu.

24

Moure la Càmera amb angles d'Euler



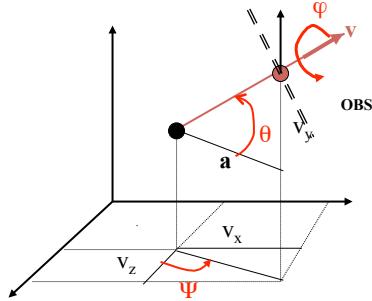
- Imaginem que en la interfície d'usuari estem ubicant la càmera movent el cursor dreta/esquerra (Ψ) i pujar/baixar (Θ). És com moure OBS sobre l'esfera i els angles d'Euler determinen un punt en esfera.
- També ho podeu pensar com si girem l'esfera per a què la seva posició respecte la càmera de defecte sigui la mateixa (agafar l'esfera amb la mà i girar-la).
- Codi per OpenGL directe a partir dels angles.



IDI 2013-2014 2Q

26

Càlcul VRP, OBS a partir angles Euler



VRP = Punt d'enfoc

OBS = VRP + d v

d > R ; per exemple: d = 2R

$$v_y = \sin(\theta); a = \cos(\theta);$$

$$v_z = \cos(\theta) \cos(\Psi);$$

$$v_x = \cos(\theta) \sin(\Psi);$$

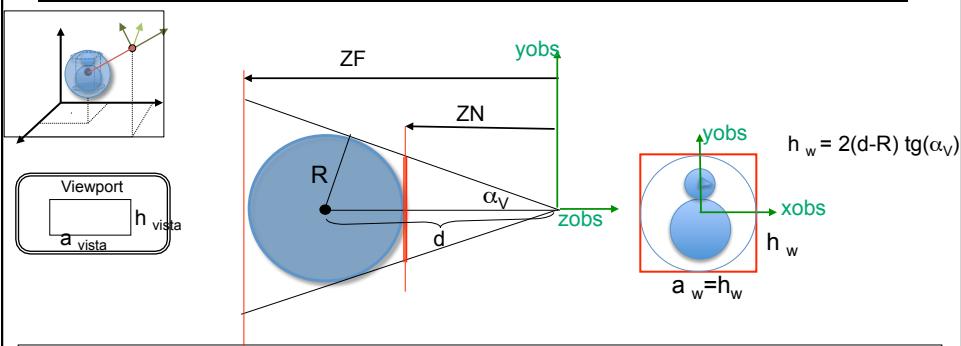
Un possible up: up = (0,1,0) ($\varphi = 0^\circ$)

Noteu que "aquí" estem considerant els angles d'orientació de la càmera ==> Ψ, θ positius quan movem la càmera cap a la dreta i quan la pugem.

IDI 2013-2014 2Q

28

Tota escena en la vista, sense deformar i càmera perspectiva



- Si tota l'esfera englobant està dins la profunditat del camp de visió, no retallem l'escena.
Per tant, $ZN \in]0, d-R]$ $ZF \in]d+R, \dots]$;
per aaprofitar precisió profunditat: $ZN = d-R$; $ZF = d+R$
- Per aaprofitar al màxim la pantalla (de fet el viewport), el window de la càmera s'ha d'ajustar a l'escena; una aproximació és ajustar el volum de visió (piràmide) de la càmera a l'esfera englobant.
 - $R = d \sin(\alpha_v)$; $\alpha_v = \text{arc sin}(R/d)$ $\Rightarrow \text{FOV} = 2\alpha_v$
 - com window està situat en ZN, α_v determina que la seva alçada sigui: $h_w = 2(d-R) \tan(\alpha_v)$
 - $ra_w = a_w/h_w = 1$ (perque α_H hauria de ser igual a α_v per asegurar que esfera no retallada)
 - Però ULL!! per a què no hi hagi deformació, cal que ra_w sigui sempre igual a ra_v , per tant, si no volem modificar el viewport:

$ra_w^* = ra_v$ amb aquesta nova ra_w^* es retallarà l'esfera? (continuarà tota dins volum de visió?)

29

Tota escena en la vista, sense deformar i càmera perspectiva

• Si $ra_v > 1$ ($>ra_w$ mínima requerida 1) => No es retalla,
no cal modificar α_v (FOV)

Amb $ra_w=1$ Amb $ra_w=ra_v$

IDI 2013-2014 2Q 30

Tota escena en la vista, sense deformar i càmera perspectiva

• Si $ra_v < 1$ ($<ra_w$ mínim requerida 1) => cal incrementar l'angle d'obertura (quedarà espai lliure a dalt i a baix)
 $FOV=2 \alpha^*_v$ on $\alpha^*_v = \text{arctg}(\text{tg}(\alpha_v) / ra_v)$

Amb $ra_w=1$ • Amb $ra_w=ra_v$ i nou FOV
 • Sempre cal calcular el nou angle a partir de l'inicial (window quadrat). Penseu que pasaria si no ho feu i modifiqueu interactivament el viewport (finestra gràfica) fent-ho >1 i <1 molts cops seguits.

31

Tota escena en la vista, sense deformar i càmera perspectiva

• Si $ra_v > 1$ ($>ra_w$ mínima requerida 1) => No es retalla, no cal modificar α_v (FOV)
Justificació: $ra_v^* = ra_w^* \cdot h_w$ serà superior a 1; si no modifiquem l'angle FOV, h_w no canvia =>
 $a_v^* = ra_w^* \cdot h_w$ i com $ra_w^* > ra_w$ => $a_v^* > a_w$ i, per tant, serà
més gran del necessari però es veurà tota l'esfera i quedarà espai pels laterals.

• Si $ra_v < 1$ ($<ra_w$ mínim requerida 1) => cal incrementar l'angle d'obertura (quedarà espai lliure a dalt i a baix)
 $FOV=2\alpha_v$ on $\alpha_v^* = \arctg(\tg(\alpha_v) / ra_v)$

Justificació: com $a_w^* = ra_w^* \cdot h_w$, si no modifiquem angle, h_w no varia; com $ra_w^* < ra_w$ => $a_w^* < a_w$ i l'esfera quedaria retallada (en horitzontal). Per tant, cal incrementar l'angle α_v (i, per tant, h_w^*) per a garantir una ampliada del window igual a la mínima requerida (igual que la h_w inicial).

- sabem: $h_w^* = a_w / ra_v = (2(d-R)\tg(\alpha_v)) / ra_v$ i per trigonometria $h_w^* = 2(d-R)\tg(\alpha_v)$
- per tant: $\alpha_v^* = \arctg(\tg(\alpha_v) / ra_v)$

32

Tota escena en la vista, sense deformar i càmera axonomètrica

• ZN i ZF mateix raonament que en càmera perspectiva.

• Window mínim requerit (centralat)= $(-R, -R, R, R)$ => una $ra_w = 1$ (per què?)

• Si $ra_w \neq ra_v$ ==> deformació (per què?)
- Si $ra_v > 1$ => cal incrementar la ra_w => modificar window
com $ra_w = a_w/h_w$ => podem incrementar a_w o decrementar h_w (és retallaria esfera!!)
Per tant:
 $a_w^* = ra_v * h_w = ra_v * 2R$ => inc_a = $a_w^* - a_w$
window = $(- (R+inc_a/2), R+inc_a/2, -R, R) = (-R\ ra_v, R\ ra_v, -R, R)$

- raonament similar per recalcular window quan $ra_v < 1$

33