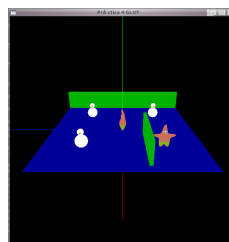
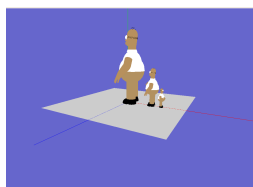


Classe 5: contingut

- Breu repàs de càmera en tercera persona i Zoom
- Moure càmera en tercera persona (mode inspecció)
- Càlcul de View Matrix amb càmera especificada amb angles d'Euler
- Exercicis

IDI 2018-2019 1Q

Càmera 3ra persona

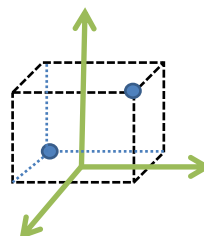


Visualització inicial de l'escena:

- inclogui tota l'escena (no retalli cap objecte)
- posició arbitrària de l'observador
- centrada en viewport
- optimitzant ocupació del viewport/vista
- sense deformació

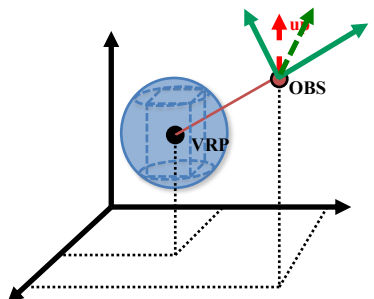
Dada: capsa mínima contenidora d'escena

$\text{cmin}=(\text{xmin}, \text{ymin}, \text{zmin})$ i $\text{cmax}=(\text{xmax}, \text{ymax}, \text{zmax})$



IDI 2018-2019 1Q

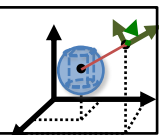
Inicialització posicionament amb OBS, VRP, up



- Centrat => **VRP=CentreEscena**
- Per assegurar que l'escena es veu sense retallar des d'una **posició arbitrària** CAL que **OBS** sempre fora capsa mínima contenidora; per assegurar-ho CAL que **OBS** fora de l'esfera englobant de la capsa => distància "d" de l'**OBS** a **VRP** superior a R esfera.
 - CapsaMinCont=(xmin,ymin,zmin,xmax,ymax,zmax)
 - CentreEscena=Centre(CapsaMinCont) = $((xmax+xmin)/2, (ymax+ymin)/2, (zmax+zmin)/2)$
 - $R = \text{dist}((xmin,ymin,zmin), (xmax,ymax,zmax))/2$
 - $d > R$; 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 homers verticals (eix Y es vegi vertical) **up**=(0,1,0)

IDI 2018-2019 1Q

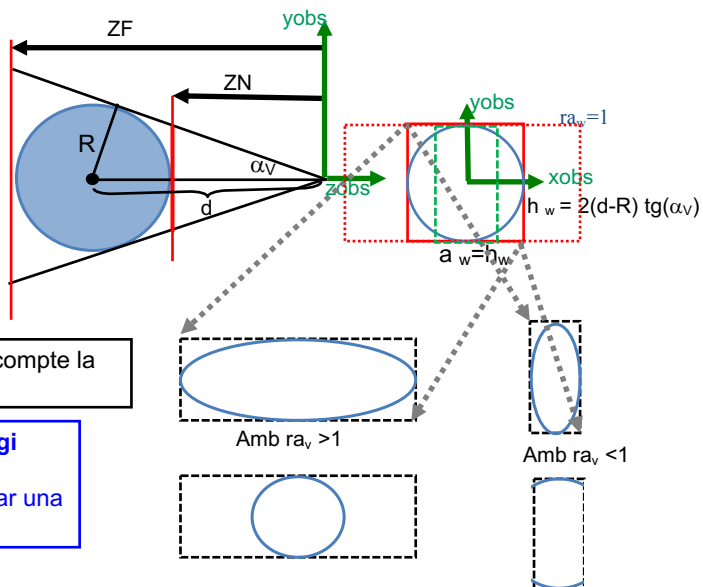
Tota escena en la vista, sense deformar i càmera perspectiva



$ZN = d - R$; $ZF = d + R$
 $\alpha_v = \arcsin(R/d)$
 $\Rightarrow FOV = 2 * \alpha_v$
 $ra_w = a_w/h_w = 1$

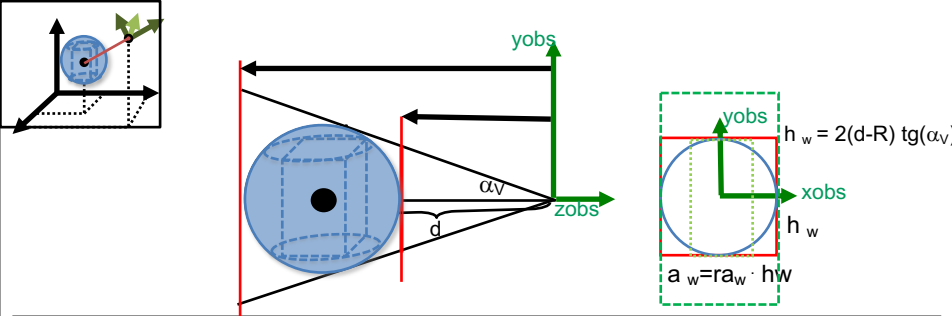
PERÒ cal tenir en compte la ra_v del viewport

Per a què **no hi hagi deformació**, cal modificar ra_w i forçar una **$ra_w^* = ra_v$**




IDI 2018-2019 1Q


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), només fer $ra_w^* = ra_v$

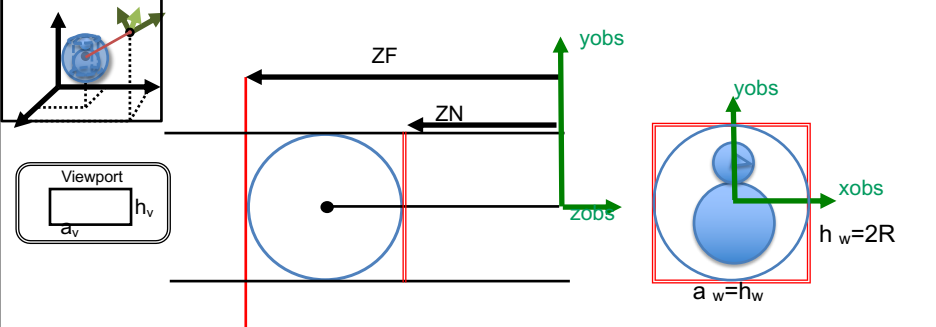


• Si $ra_v < 1$ ($< ra_w$ mínima requerida 1) => cal fer $ra_w^* = ra_v$ i incrementar l'angle d'obertura $FOV = 2 \alpha_v^*$ on $\alpha_v^* = \arctg(tg(\alpha_v) / ra_v)$



IDI 2018-2019 1Q

Tota escena en la vista, sense deformar i càmera ortogonal



• ZN i ZF mateix raonament que en càmera perspectiva.

• Window mínim requerit (centrat) = $(-R, R, -R, R)$ => una $ra_w = 1$

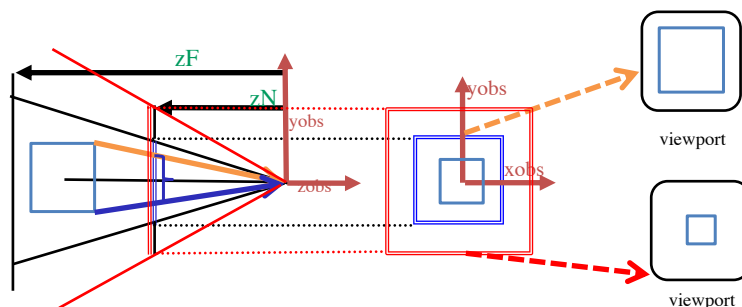
• Si $ra_w \neq ra_v$ ==> deformació

- 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, modifiquem a_w :
 $a_w^* = ra_v * h_w = ra_v^2 * R$
 window = $(-R * ra_v, R * ra_v, -R, R)$
- Raonament similar per recalculer window quan $ra_v < 1$

IDI 2018-2019 1Q

L'òptica i el ZOOM

- Modificar l'angle d'obertura (tot mantenint la ra) ;
 Modificar **window** en **ortogonal**
- Modificar la distància de l'Obs al VRP
 (modificant ZN i ZF adequadament)
- Modificar Obs i VRP en la direcció $-v \rightarrow$ travelling



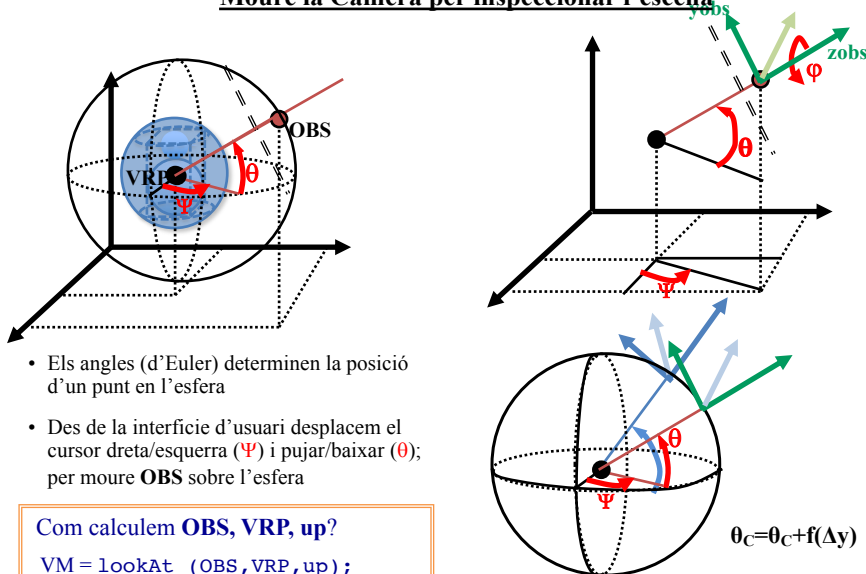
IDI 2018-2019 1Q

Classe 5: contingut

- Breu repàs de càmera en tercera persona i Zoom
- Moure càmera en tercera persona (mode inspecció)
- Càlcul de View Matrix amb càmera especificada amb angles d'Euler
- Exercicis

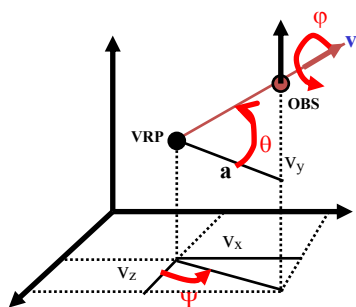
IDI 2018-2019 1Q

Moure la Càmera per inspeccionar l'escena



IDI 2018-2019 1Q

Càlcul VRP, OBS a partir dels angles d'Euler



VRP = Punt d'enfoc

OBS = $\text{VRP} + d \mathbf{v}$

$d > R$; 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**: $\mathbf{up} = (0, 1, 0)$ ($\phi = 0^\circ$)

Noteu que l'òptica no cal modificar-la perquè ens movem sobre esfera que envolta l'escena de radi d .

Es podria calcular la View Matrix directament a partir dels angles?

Noteu que estem considerant els angles d'orientació de la càmera:

Ψ en $[-180, 180]$, θ en $[-90, 90]$

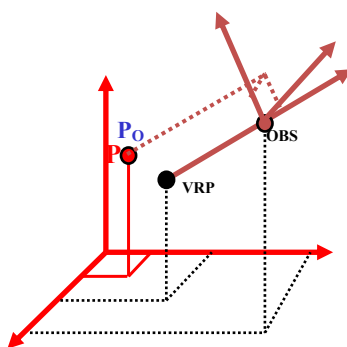
positius quan movem la càmera cap \rightarrow i quan la movem cap \uparrow

Classe 5: contingut

- Breu repàs de càmera en tercera persona i Zoom
- Moure càmera en tercera persona (mode inspecció)
- **Càlcul de View Matrix amb càmera especificada amb angles d'Euler**
- Exercicis

IDI 2018-2019 1Q

Càlcul view Matrix directe a partir d'angles Euler, VRP i d

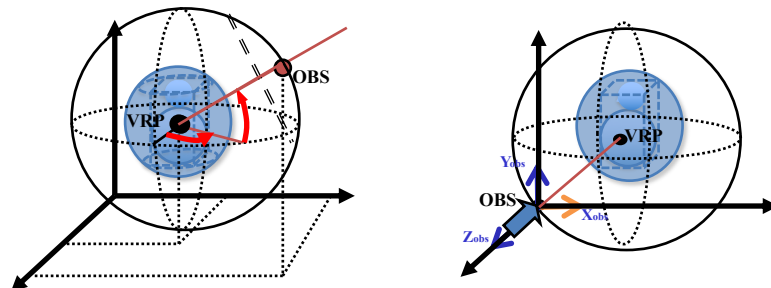


RECORDEU:

La viewMatrix serveix per tenir posició de punts respecte observador

IDI 2018-2019 1Q

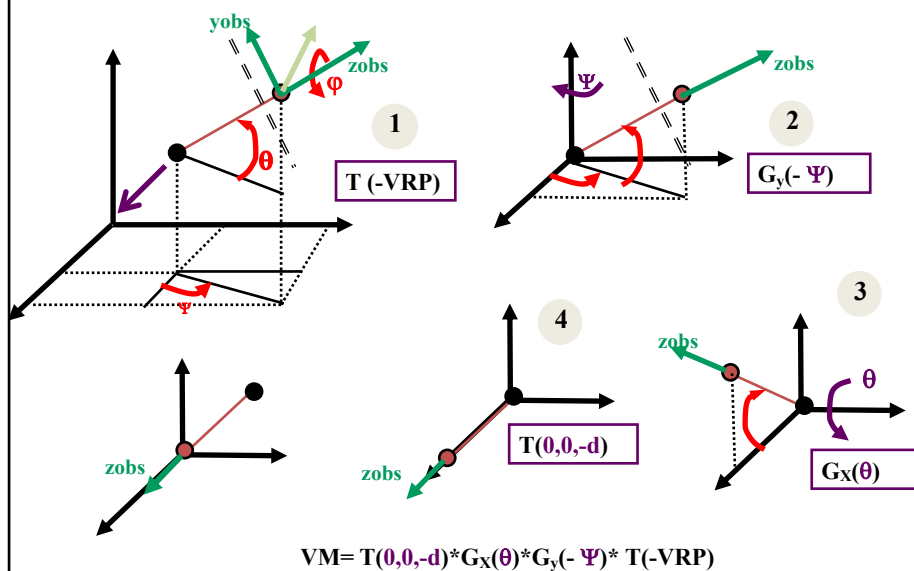
Càlcul VM directe a partir d'angles Euler, VRP i d



- 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 i posicionar-la.
- Noteu que zobs passarà a ser coincident amb zA (SCO i SCA coincidiran)
- **Pensem el moviment tenint en compte que sabem calcular matrius de gir només si girem entorn d'eixos que passen per origen de coordenades.**

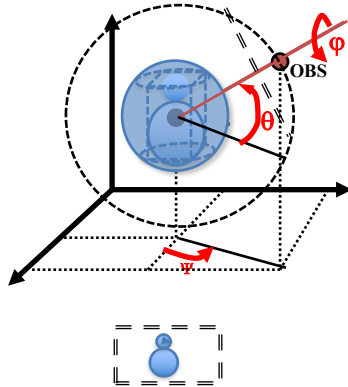
IDI 2018-2019 1Q

Càlcul MV directe a partir d'angles Euler: exemple més complex



IDI 2018-2019 1Q

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



$$VM = T(0,0,-d) * G_Z(-\varphi) * G_X(\theta) * G_Y(-\Psi) * T(-VRP)$$

```
VM=Translate (0,0,-d)
VM=VM*Rotate(-φ,0,0,1)
VM= VM*Rotate (θ,1,0,0.)
VM= VM*Rotate(-ψ,0,1,0.)
VM= VM*Translate(-VRP.x,-VRP.y,-VRP.z)
viewMatrix[VM]
```

Compta 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.

IDI 2018-2019 1Q

```
(1)/* CreateBuffers(); Crear VAO del model
(un cop)*/
(2) // IniCamera() calcular paràmetres càmera inicial
(3) /*viewTransform() amb lookAt o TG d'Euler per
calcular la VM inicial i cada cop que es modifica
algún paràmetre de posició de la càmera*/
VM = lookAt(OBS,VRP,UP);
viewMatrix[VM];
(4)/*projectTransform() per calcular PM per
òptica inicial i cada cop que es modifica
algún dels seus paràmetres*/
PM=perspective (FOV,ra,zN,ZF);
projectMatrix[PM];
(5)/*resize(...) Per garantir no deformació; ull
que afectarà parametres d'òptica*/
(6) /*PaintGL(); cada cop que es requereix
refresc de la visualització*/
/*per cada instància d'un model:
modelTransform() Calcula TGi i passar a OpenGL*/
modelTransform_i(TG);
modelMatrix(TG);
Pinta_model(VAO);
```

Recordatori inicialitzacions per visualització

Vertex Shader

```
in vec3 vertex;
uniform mat4 TG, VM, PM;
void main ()
{
    gl_Position =
        PM*VM*TG*vec4(vertex,1.0);
}
```

IDI 2018-2019 1Q