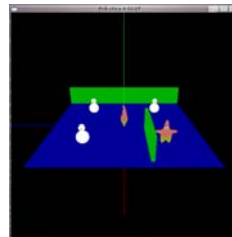
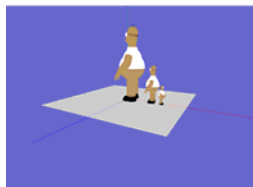


Classe 5: contingut

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

IDI Q2 2019-2020

Vist: Càmera 3ra persona

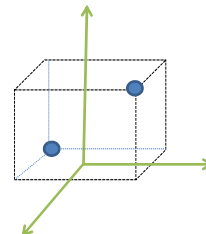


Càmera per a què la visualització inicial de l'escena:

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

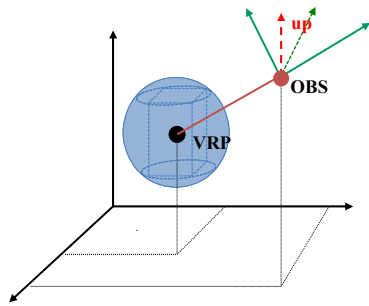
Dada: capsa mínima contenidora de l'escena

$cmin=(xmin, ymin, zmin)$ i $cmax=(xmax, ymax, zmax)$



IDI Q2 2019-2020

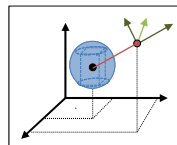
Vist: Inicialització posicionament amb OBS, VRP, up



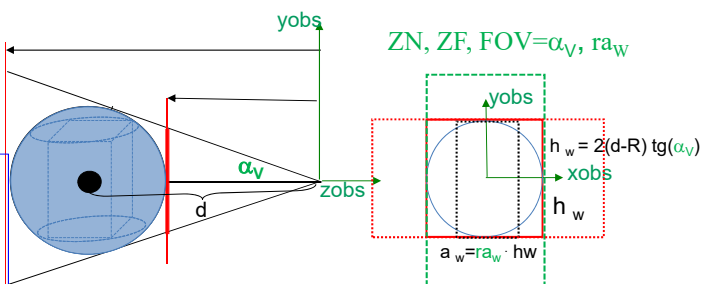
- Centrat => **VRP=CentreEscena**
- Per assegurar que l'escena es veu sense retallar des d'una **posició arbitrària** CAL que **OBS** sempre fora capsa mínima contenidora; per assegurar-ho CAL que **OBS** fora de l'esfera englobant de la capsa => distància "d" de l'**OBS** a **VRP** superior a R esfera.
 - 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** + $d \cdot \mathbf{v}$; \mathbf{v} normalitzat en qualsevol direcció;
per exemple $\mathbf{v} = (1,1,1) / \|(1,1,1)\|$
- **up** qualsevol que no sigui paral·lel a \mathbf{v} ; si volem homers verticals (que l'eix Y_A es vegi vertical)
up=(0,1,0)

IDI Q2 2019-2020

Vist: òptica perspectiva per veure tota l'escena i $ra_w = ra_v$



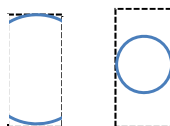
$ZN = d - R$; $ZF = d + R$
 $\alpha_v = \arcsin(R/d)$
 $\Rightarrow FOV = 2 \cdot \alpha_v$
 $ra_w = 1$; però fem $ra_w^* = ra_v$



- Si $ra_v > 1$ (és > que la ra_w mínima requerida que és 1) => No es retalla l'escena al fer **$ra_w^* = ra_v$** , no cal modificar α_v (FOV)



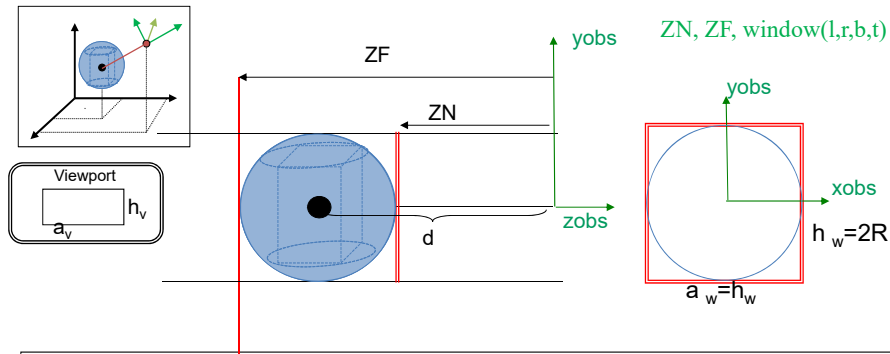
- Si $ra_v < 1$ (és < que la ra_w mínima requerida que és 1) => al fer **$ra_w^* = ra_v$** es retalla l'escena



Cal incrementar l'angle FOV d'obertura
 $FOV^* = 2 \alpha_v^*$ on
 $\alpha_v^* = \arctg(\tg(\alpha_v) / ra_v)$

IDI Q2 2019-2020

Vist: òptica ortogonal per veure tota l'escena i $ra_w = ra_v$



- **ZN i ZF** mateix raonament que en càmera perspectiva.
- **Window mínim requerit (centrat) = $(-R, R, -R, R)$ \Rightarrow una $ra_w = 1$ (per què ?)**
- Si $ra_w \neq ra_v \Rightarrow$ deformació (per què?)
 - Si $ra_v > 1 \Rightarrow$ cal incrementar la $ra_w \Rightarrow$ *modificar window*
 com $ra_w = a_w/h_w \Rightarrow$ podem incrementar a_w o decrementar h_w (és retallaria l'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 $ra_v < 1$

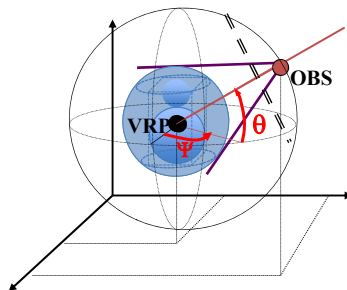
IDI Q2 2019-2020

Classe 5: contingut

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

IDI Q2 2019-2020

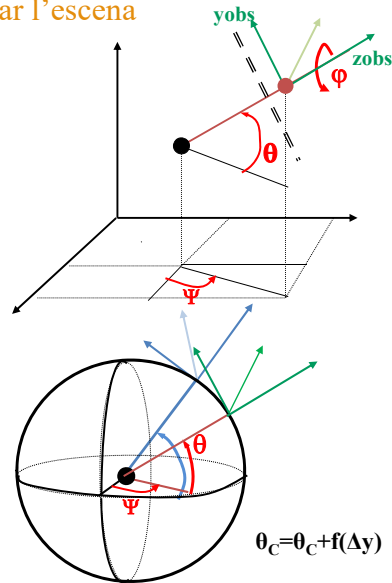
Moure la Càmera per inspeccionar l'escena



- Els angles (d'Euler) determinen la posició d'un punt sobre l'esfera
- Des de la interfície d'usuari desplaçem el cursor dreta/esquerra (Ψ) i pujar/baixar (θ); per moure **OBS** sobre l'esfera
- **NOTEU: que l'òptica no canvia**

Com calculem **OBS**, **VRP**, **up**?

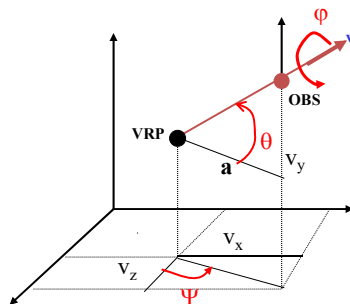
```
VM = lookAt (OBS, VRP, up);  
viewMatrix (VM);
```



$$\theta_c = \theta_c + f(\Delta y)$$

IDI Q2 2019-2020

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**: **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 ViewMatrix directament a partir dels angles?

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

$$\Psi \text{ en } [-180, 180], \theta \text{ en } [-90, 90]$$

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

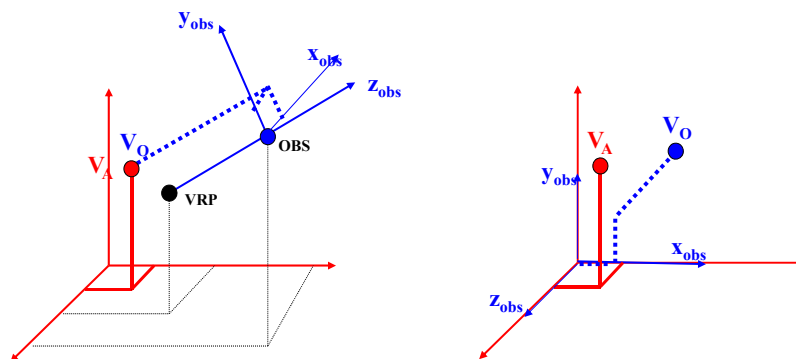
IDI Q2 2019-2020

Classe 5: contingut

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

IDI Q2 2019-2020

Càlcul viewMatrix directe a partir d'angles Euler, VRP i d (1)



RECORDEU: La viewMatrix serveix per tenir la posició dels vèrtexs respecte del SCO

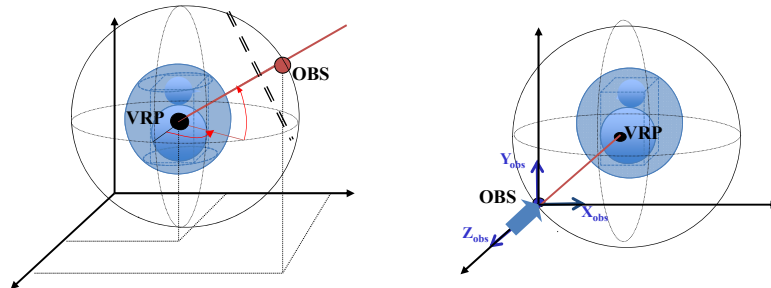
$$V_o = VM * V_A$$

Es pot calcular la VM:

- Pensant que movem la càmera (OBS, VRP, Up)
- Pensant que tenim una càmera fixa al SCA i ubiquem tota l'escena respecte d'ella → realitzar una mateixa TG_{VM} a tots els objectes. Si vèrtexs queden respecte a la càmera en la mateixa posició → TG_{VM} serà igual a la VM calculada amb OBS, VRP, Up

IDI Q2 2019-2020

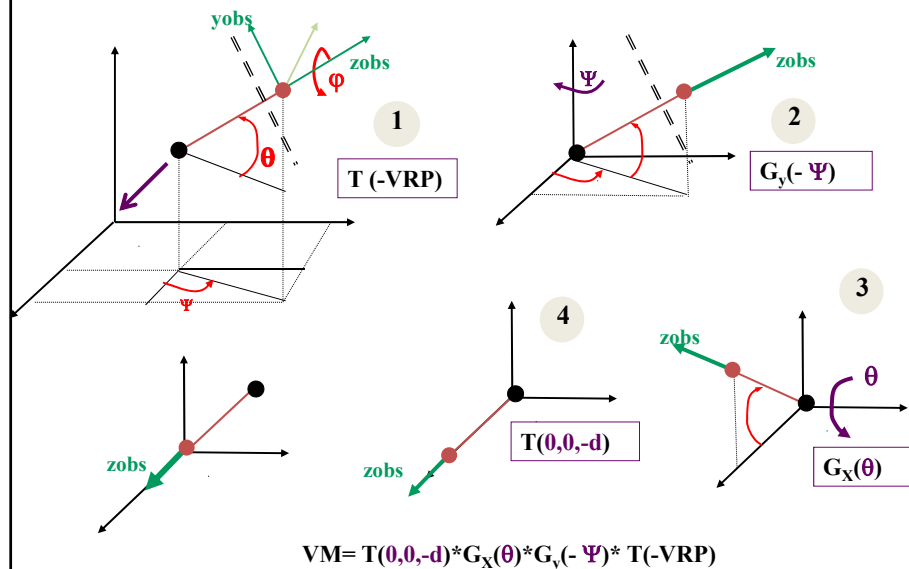
Càlcul VM directe a partir d'angles Euler, VRP i d (2)



- 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.**

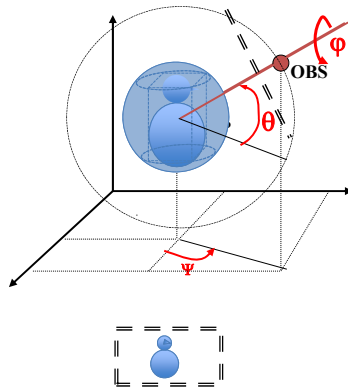
IDI Q2 2019-2020

Càlcul MV directe a partir d'angles Euler (3)



IDI Q2 2019-2020

Exemple: Posicionament amb angles Euler (4)



$$VM = T(0,0,-d) * G_z(-\varphi) * G_x(\theta) * G_y(-\Psi) * T(-VRP)$$

```
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(-VRPx,-VRPy,-VRPz)
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.

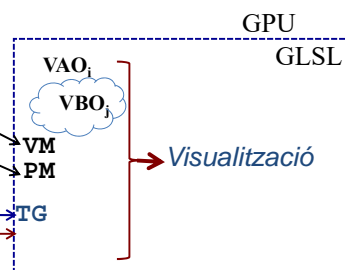
IDI Q2 2019-2020

Resum final: com visualitzar l'escena?

Calcular i enviar viewMatrix;

Calcular i enviar projectMatrix

```
per cada objectei
modelTransformi(TGi);
modelMatrix(TGi);
pinta_modeli();
fper
```



```
#version 330 core
in vec3 vertex;
uniform mat4 PM;
uniform mat4 VM;
uniform mat4 TG;
void main() {
    gl_Position = PM*VM*TG*vec4 (vertex, 1.0);
}
```

Vertex Shader

```
#version 330 core
out vec4 FragColor;

void main() {
    FragColor = vec4(0, 0, 0, 1);
}
```

Fragment Shader

IDI Q2 2019-2020

Classe 5: conceptes i preguntes

- Metàfora d'inspecció d'una escena movent el cursor
- Angles d'Euler
- Càlcul de la viewMatrix concatenant transformacions geomètriques.
- Pseudocodi per crear la matriu anterior. Importa l'ordre de les transformacions?
- Inicialització d'angles per veure l'escena en planta i pseudo-codi de càlcul de la viewMatrix.
- Quines matrius cal tornar a enviar a la GPU si es fa un “resize”?
- Si per teclat indiquem que volem moure un objecte fent una translació de (5,0,0), què cal recalcular? quina informació cal modificar de la que s'envia a la GPU?
- Exercicis proposats per donats uns OBS, VRP i up, calcular la viewMatrix amb transformacions geomètriques (i viceversa)

IDI Q2 2019-2020