

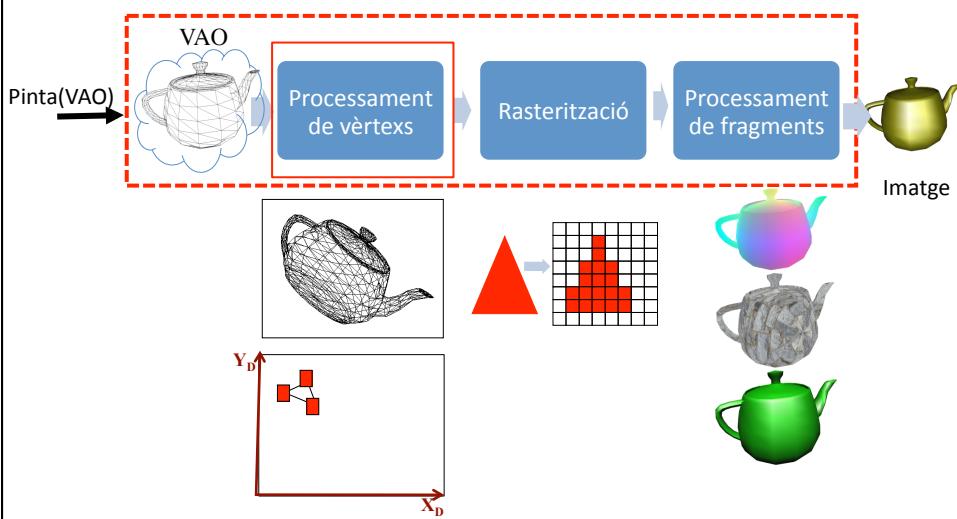
Classe 3: contingut

- Visualització: breu repàs.
- Càmera (1)
 - Posicionament: OBS, VRP, up
 - Òptica perspectiva: definició i paràmetres
 - Exemple
- Visualització: processat vèrtexs (2 i fi)

IDI Q1 2016-2017

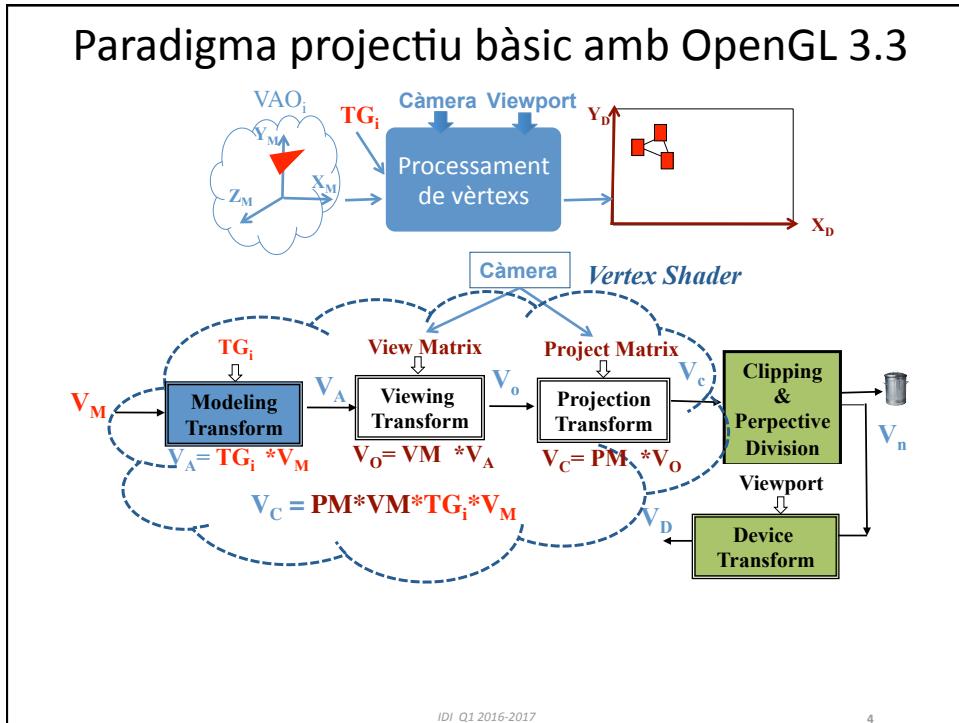
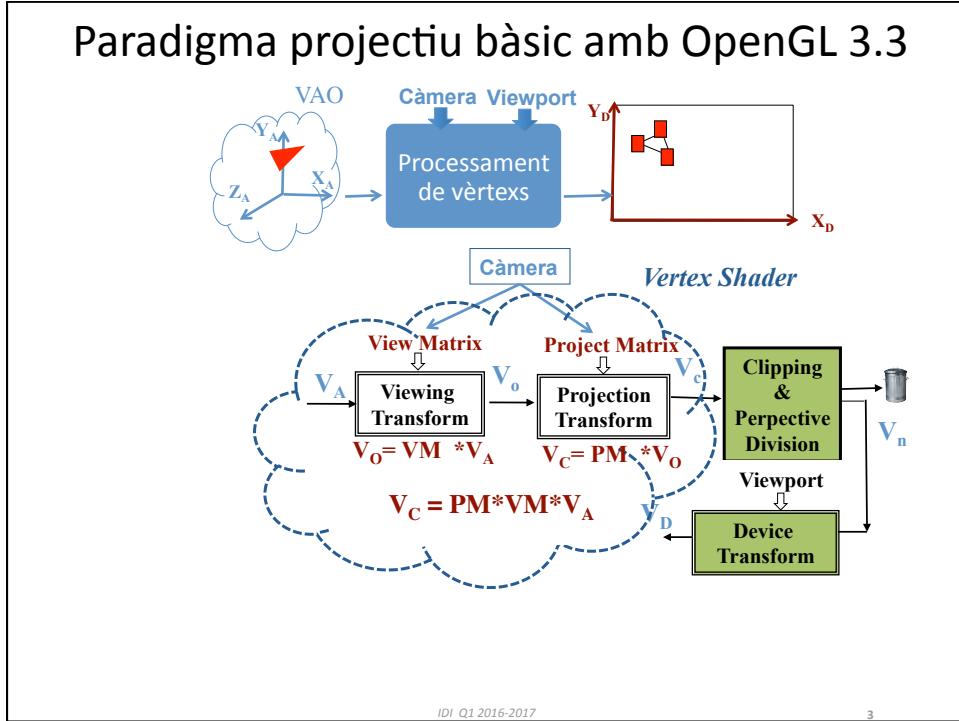
1

Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

2



Posicionament de la càmera (1): OBS, VRP, up

OBS = Observador
VRP = View Reference Point
up = View Up Vector
up "indica" la direcció de l'eix vertical de la Càmera (inclinació)

Per a la determinació dels seus valors suposem que tots els objectes de l'escena estan al seu lloc ➔ en SCA

IDI Q1 2016-2017

5

OBS, VRP, up → Càcul de la View Matrix

$$\text{VM} = \begin{bmatrix} s.x & s.y & s.z & 0 \\ w.x & w.y & w.z & 0 \\ F.x & F.y & F.z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \text{Trans}(-\text{OBS})$$

$\mathbf{F} = \mathbf{OBS} - \mathbf{VRP} = (F.x, F.y, F.z) \quad \mathbf{F} = \mathbf{F} / \| \mathbf{F} \|$
 $\mathbf{s} = \mathbf{up} \times \mathbf{F} \quad \mathbf{s} = \mathbf{s} / \| \mathbf{s} \|$
 $\mathbf{w} = \mathbf{F} \times \mathbf{s}$

```

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

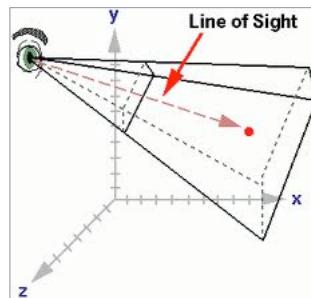
```

IDI Q1 2016-2017

6



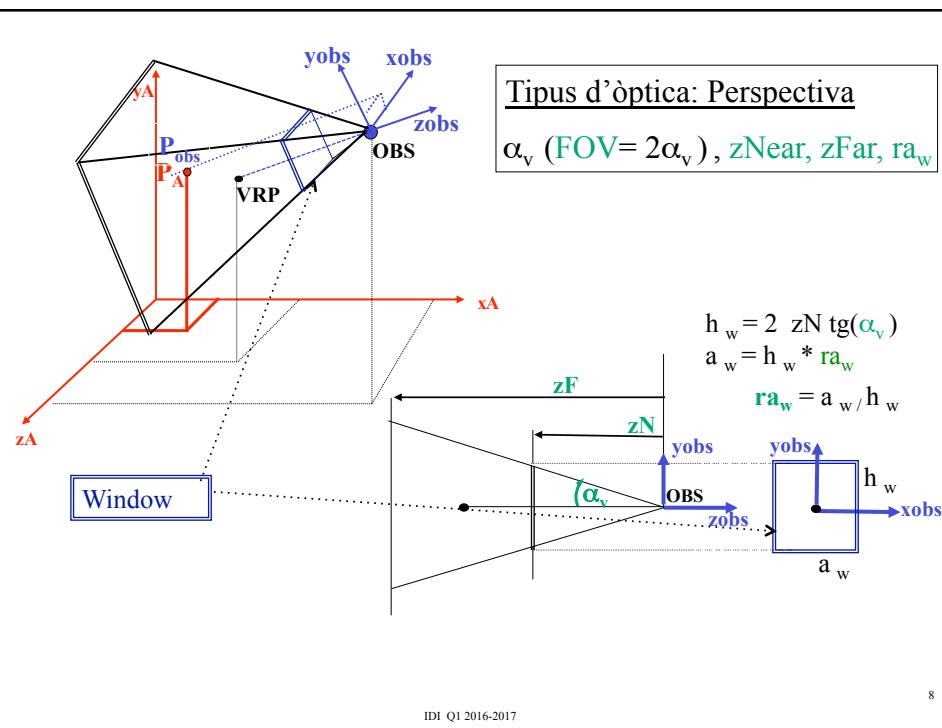
Òptica de la càmera (1): perspectiva



- Determina geometria potencialment visible
- **EL VOLUM de VISIÓ -ÒPTICA- SEMPRE es defineix RESPECTE L'OBSERVADOR**

IDI Q1 2016-2017

7



FOV, zNear, zFar, ra → Càcul de matriu de projecció (PM)

FOV, zNear, zFar, ra_w

$$PM = \begin{pmatrix} 1/ra*a & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad \begin{array}{l} a = \tan(FOV/2) \\ c = (f+n)/(n-f) \\ d = 2nf/(n-f) \end{array}$$



$$V_c = PM * V_o = (x_c, y_c, z_c, w)$$

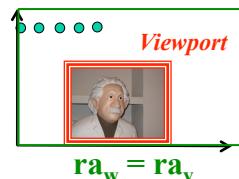
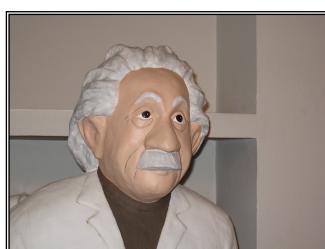
```
PM=perspective (FOV,ra,zN,ZF);
projectMatrix(PM);
```

IDI Q1 2016-2017

9

Sobre la relació d'aspecte del window i del viewport

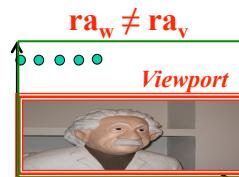
Window



$$ra_w = ra_v$$

Per a no tenir deformació en la imatge

$$ra_w = ra_v$$



$$ra_w \neq ra_v$$

IDI Q1 2016-2017

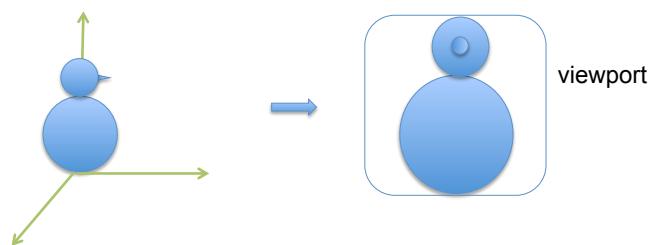
10

Exemple 1: Donades les funcions:

- `pinta_esfera()` que envia a pintar una esfera de radi 1 i centre en (0,0,0)
- `pinta_con()` que envia a pintar un con amb base centrada en origen i radi 1, i alçada 1 amb direcció del seu eix segons Z^+

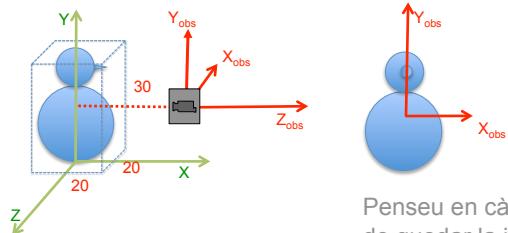
Indica:

- El pseudo-codi d'una funció `pinta_ninot()` que pintaria un objecte com el de la figura, format per: una esfera de radi 10 i centre (0,10,0), una 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^+
- tots els paràmetres d'una càmera que permeti obtenir la imatge similar a la que s'indica, en un viewport de 600x600 que ocupa tota la finestra gràfica.



IDI Q1 2016-2017

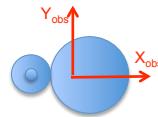
11



Penseu en càmera i com ha de quedar la imatge

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

```
VRP=(0,15,0)
OBS=(30,15,0)
Up=(0,1,0)
```



Quins paràmetres si volem que quedí així?

IDI Q1 2016-2017

12

Òptica de la càmera: Determina el Volum de Visió

1. Posició, orientació
2. Òptica

viewport

IDI Q1 2016-2017 13

Exemple 1: Òptica perspectiva

VRP=(0,15,0); OBS=(30,15,0), up=(0,1,0)

600
600
viewport

$z_N = 20; z_F = 40$
 $\alpha = \arctg(15/20) \rightarrow \alpha = 36,8^\circ$
 $r_a = 20/30 = 0,66$
 Com $r_a = 1 \rightarrow$ deformació
 Solució $r_a = 1$

IDI Q1 2016-2017 14

```

// Crear VAOs de models (un cop);
...
/* calcular paràmetres càmera i
matrius cada cop que es
modifiquin */
VM = lookAt(OBS,VRP,UP);
viewMatrix(VM);
PM=perspective (FOV,ra,zN,ZF);
projectMatrix(PM);
glViewport (0,0,w.h);
...
// cada cop que es requereix refresc
per cada objecte
//obtenir TGi i passar a OpenGL
...
modelMatrix (TGi);
pinta_model (VAOi);
fper

```

Vertex Shader

```

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

```

IDI Q1 2016-2017

15

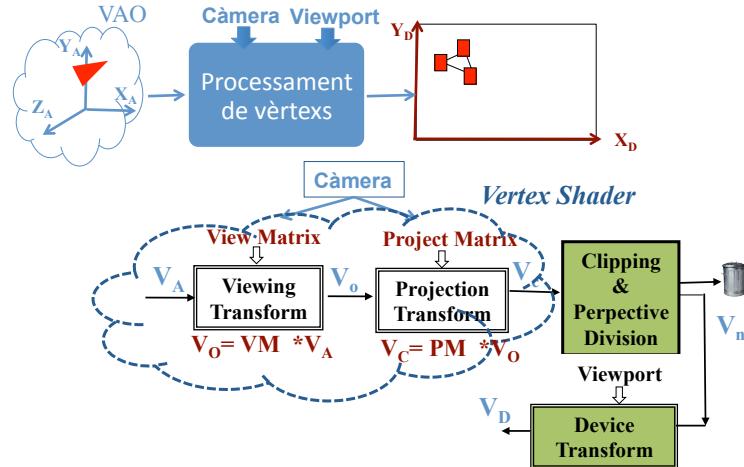
Classe 3: contigut

- Visualització: breu repàs.
- Càmera (1)
 - Posicionament: OBS, VRP, up
 - Òptica perspectiva: definició i paràmetres
 - Exemple
- Visualització: processat vèrtexs (2 i fi)

IDI Q1 2016-2017

16

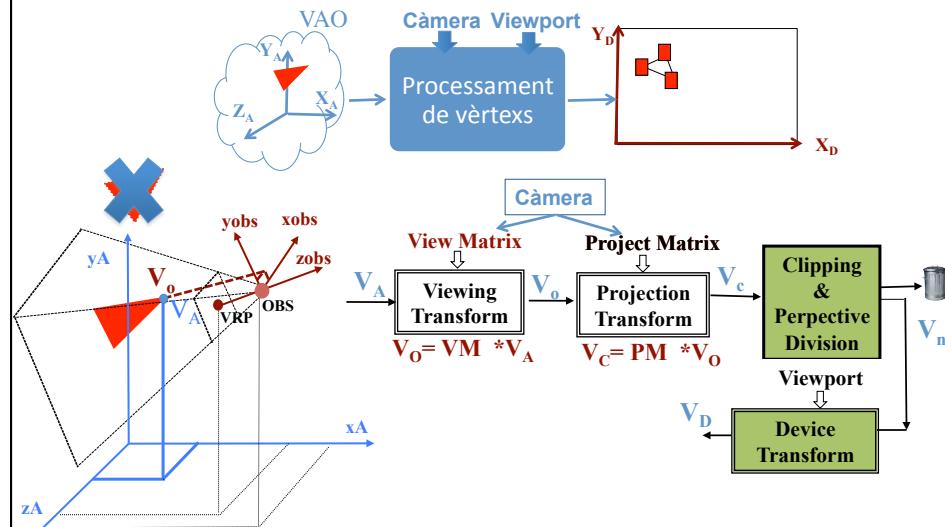
Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

17

Paradigma projectiu bàsic amb OpenGL 3.3



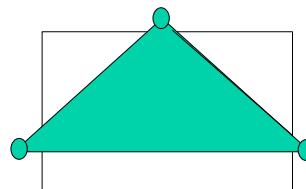
IDI Q1 2016-2017

18

Condició per a que un Vèrtex sigui interior al volum de visió:

$$\begin{aligned} -W_c &\leq x_c \leq W_c \\ -W_c &\leq y_c \leq W_c \\ -W_c &\leq z_c \leq W_c \end{aligned}$$

$$W_c = -Z_o$$



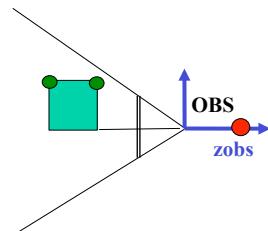
IDI Q1 2016-2017

19

$$\begin{array}{l} \text{FOV}=90^\circ \\ \text{ra=1} \\ \text{n=1, f=11} \end{array}$$

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1.2 & -2.2 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1.2 & -2.2 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ -1.2z_e - 2.2 \\ -z_e \end{bmatrix}$$



Exemple: $(0, 1, -2, 1)_o \Rightarrow (0, 1, 0.2, +2)_c$

Exemple: $(0, 1, -3, 1)_o \Rightarrow (0, 1, 1.4, +3)_c$

Exemple: $(0, 0, 1, 1)_o \Rightarrow (0, 0, -3.4, -1)_c$

IDI Q1 2016-2017

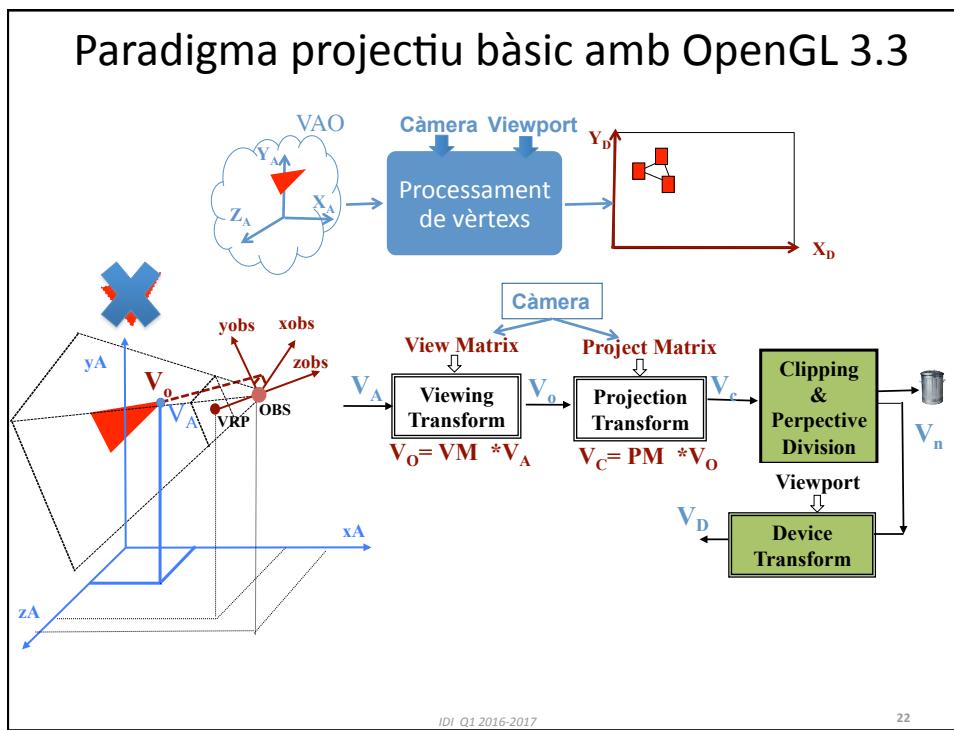
20

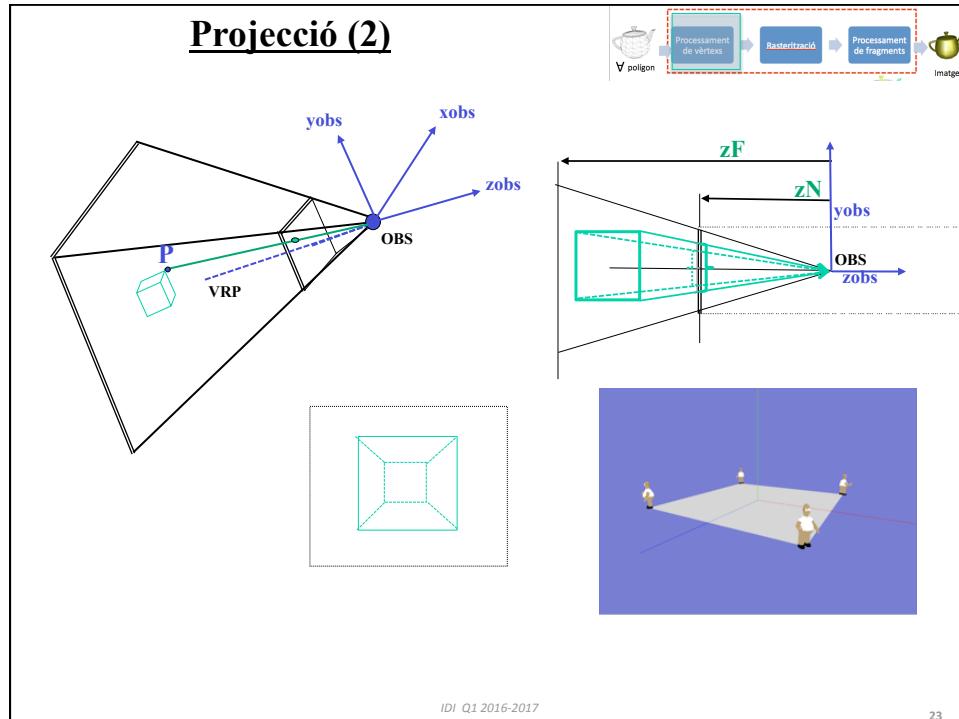
Condició per a que un Vèrtex sigui interior al volum de visió:

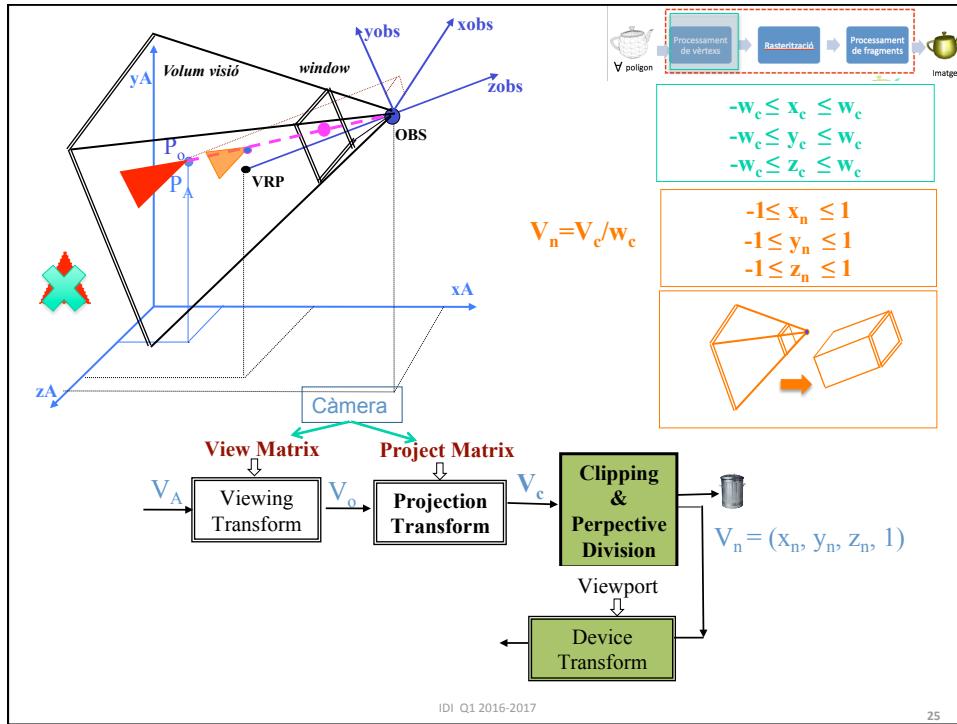
$$\begin{aligned} -W_c &\leq x_c \leq W_c \\ -W_c &\leq y_c \leq W_c \\ -W_c &\leq z_c \leq W_c \\ W_c = -Z_o \end{aligned}$$

IDI Q1 2016-2017

21







Given parameters: $\text{FOV} = 90^\circ$, $\text{ra} = 1$, $n = 1$, $f = 11$. These are converted into a projection matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1.2 & -2.2 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

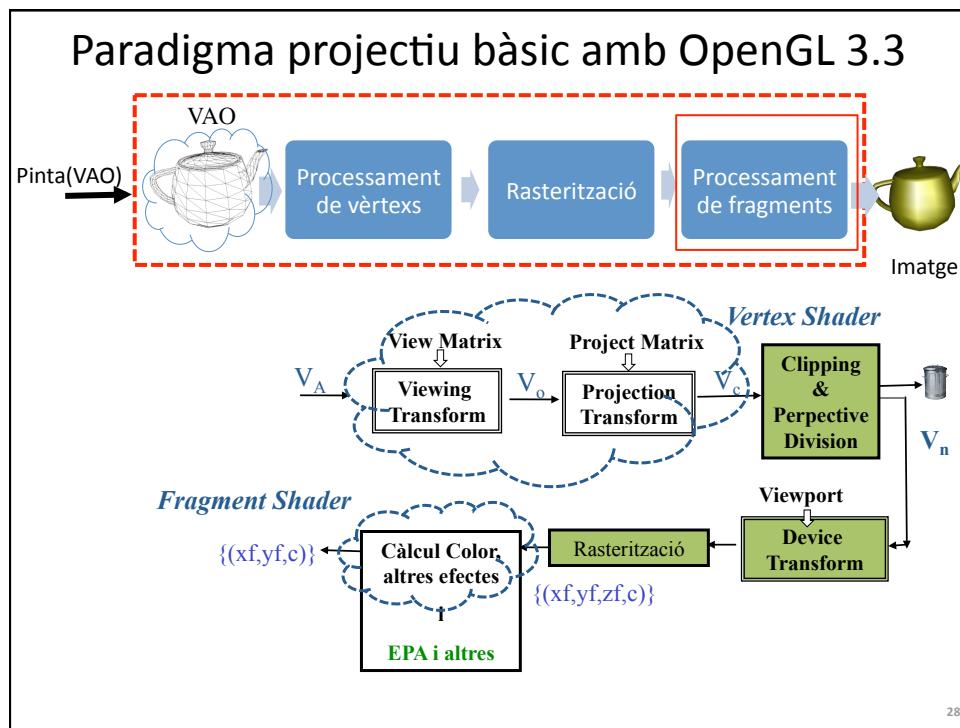
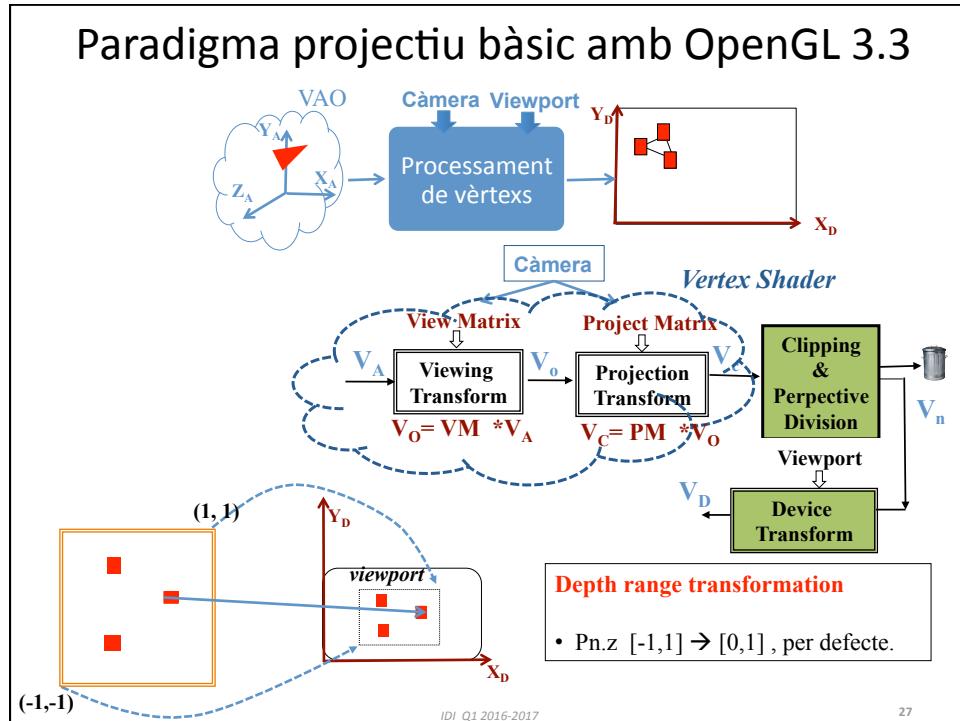
The eye coordinates $\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix}$ are multiplied by this matrix to get the normalized device coordinates $\begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ -1.2z_e - 2.2 \\ -z_e \end{bmatrix}$.

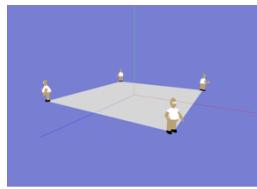
Exemple: $(0, 1, -2, 1)_o \Rightarrow (0, 1, 0.2, +2)_c \Rightarrow (0, 0.5, 0.1)$

Exemple: $(0, 1, -3)_o \Rightarrow (0, 1, 1.4, +3)_c \Rightarrow (0, 0.33, 0.4)$

IDI Q1 2016-2017

26





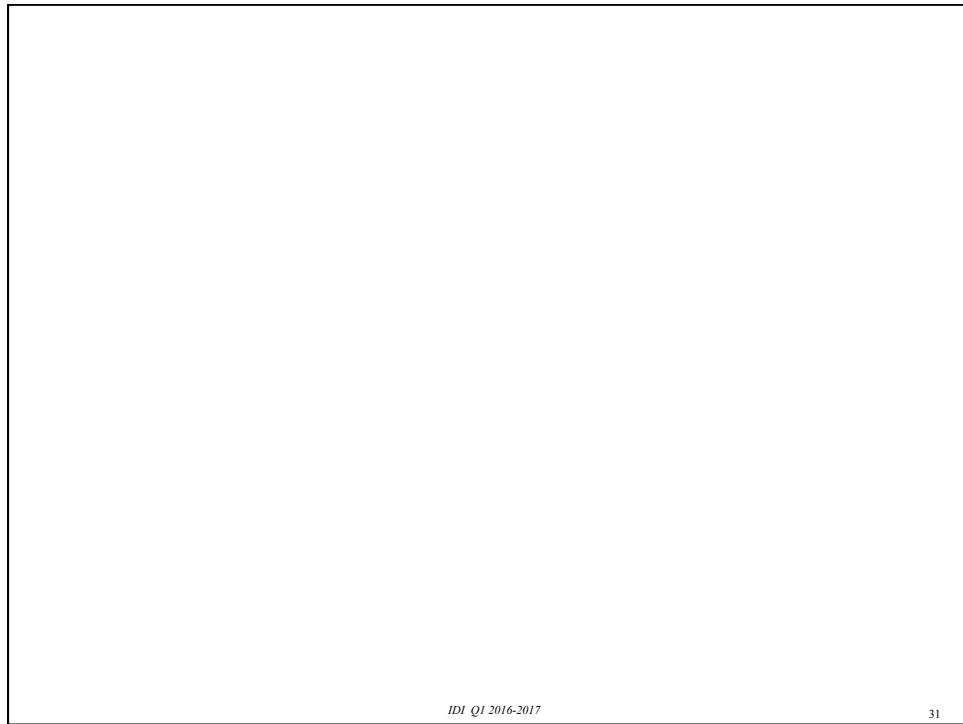
Per pensar...

Quins podrien ser uns **paràmetres de posició, orientació i òptica** per a una càmera que, donada una escena i coneguda la seva capsa mínima contenidora

(x_{\min} , y_{\min} , z_{\min}) - (x_{\max} , y_{\max} , z_{\max}), visualitzi una imatge que inclogui totalment l'escena, ocupant el màxim de la vista (viewport) i sense deformació?

Per pensar...

- Com fer un zoom?
- Quins paràmetres de càmera modificaries?



IDI Q1 2016-2017

31