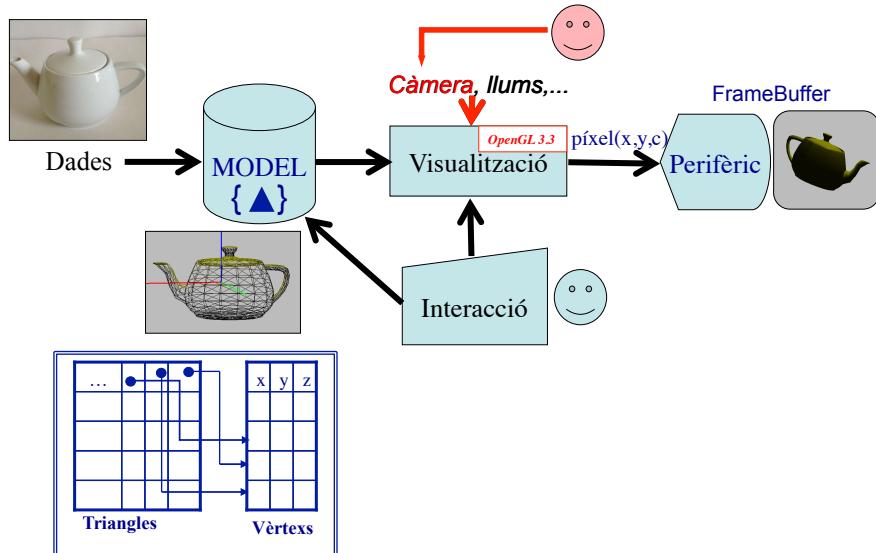


Classe 3: contingut



IDI 2018-2019 1Q

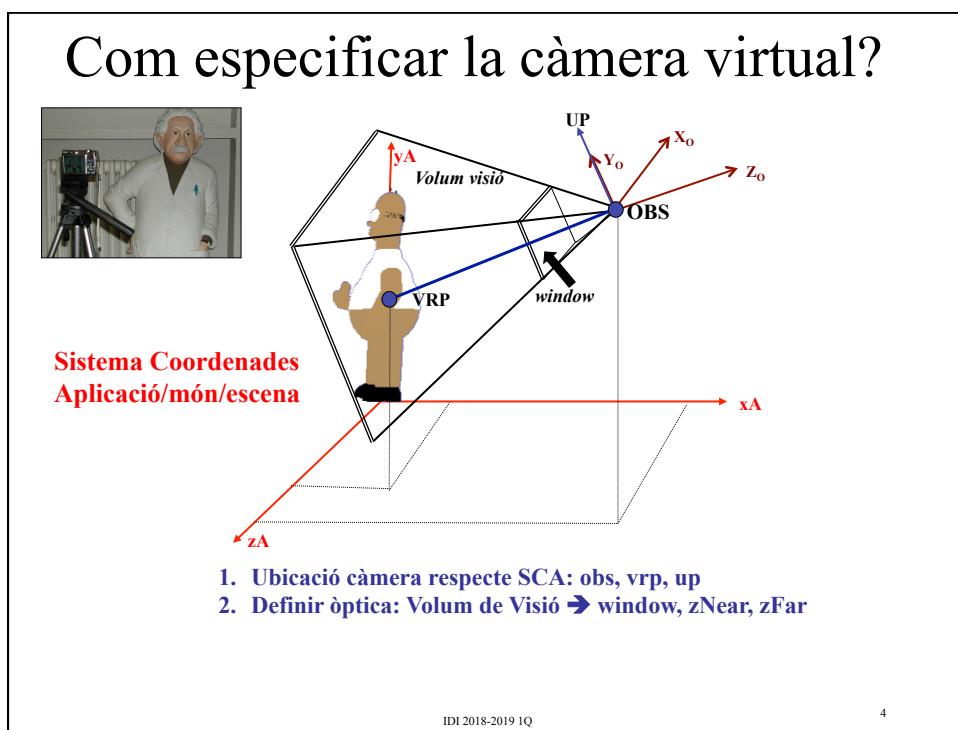
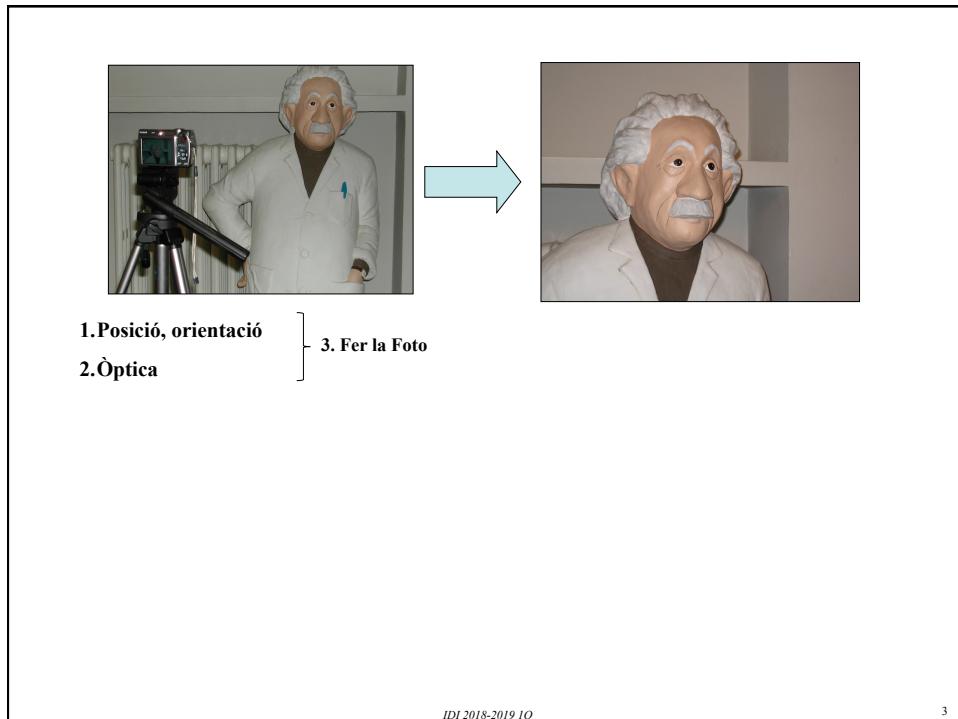
1

Classe 3: contingut

- Especificació de càmera i “viewport”
- El procés de visualització projectiu
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments
 - El fragment shader
- Exemples i primers exercicis

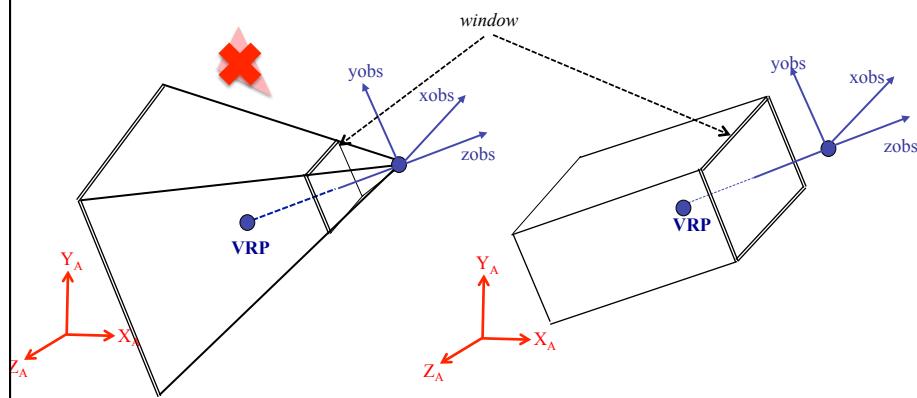
IDI 2018-2019 1Q

2





Òptica: perspectiva o ortogonal



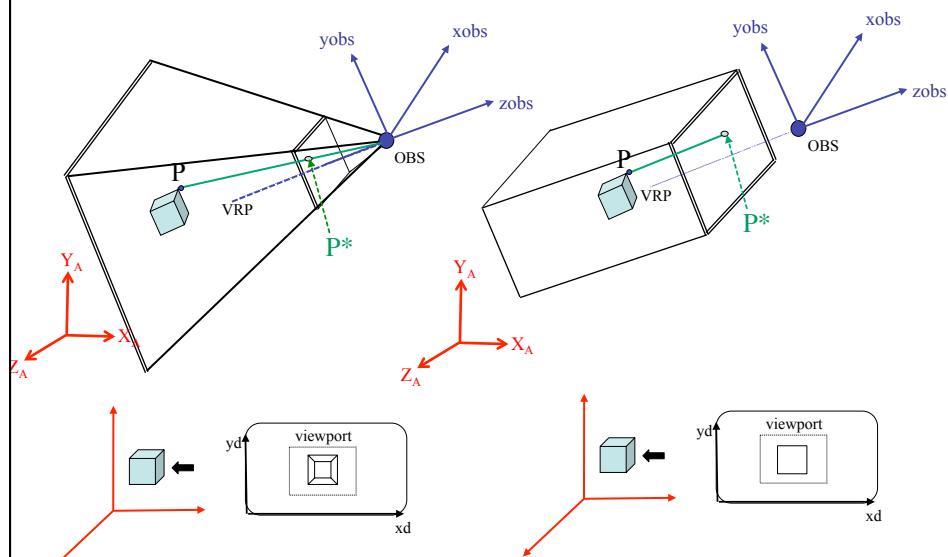
Definir òptica: volum de visió \rightarrow (window, zNear, zFar)

IDI 2018-2019 IQ

5

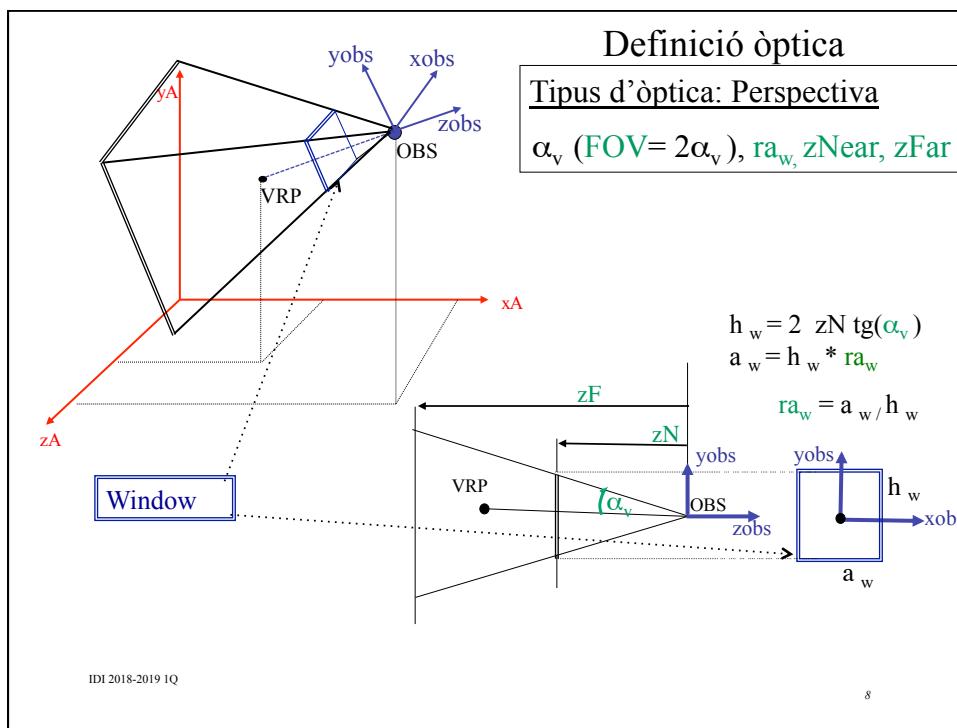
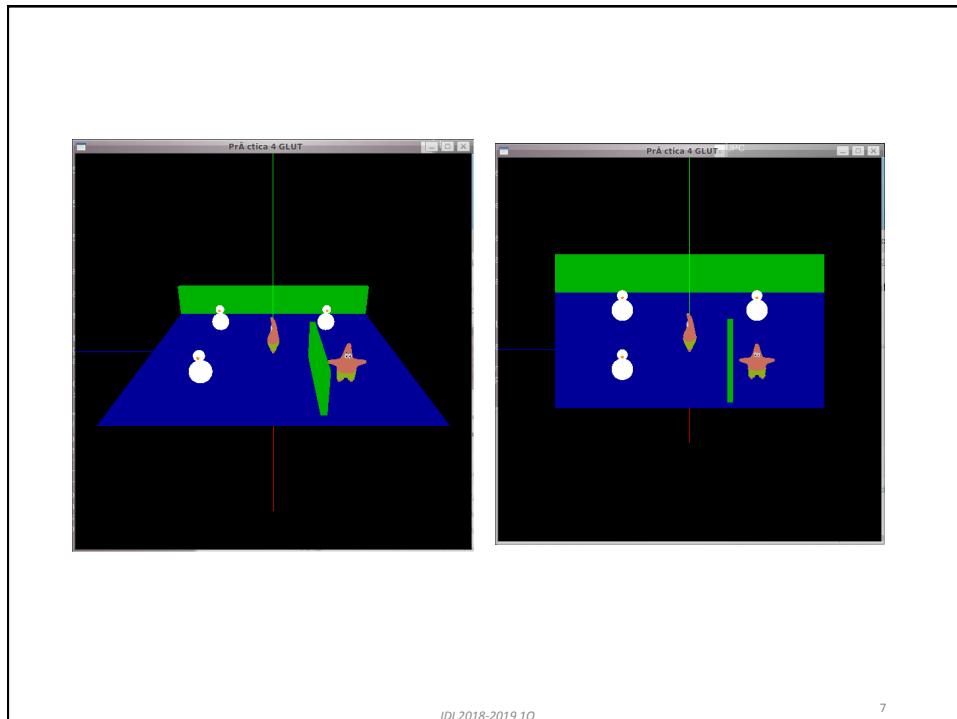


Fer la foto: perspectiva/ortogonal



IDI 2018-2019 IQ

6



Definició òptica

The diagram illustrates the optical definition of a camera frustum. It shows a 3D scene with a camera at point OBS looking along the z_{obs} axis. The frustum is defined by the near plane (z_{Near}) and the far plane (z_{Far}). A vertical red line represents the optical axis (yA). A horizontal red line represents the image plane. A box labeled "Window" indicates the area of interest on the image plane. A coordinate system (l, b) is shown at the bottom left of the image plane, with axes x_{obs} and y_{obs} . A point VRP (View Ray Point) is shown on the z_{obs} axis. A right-angled triangle is drawn on the image plane to show the relationship between the window coordinates (l, b) and the normalized device coordinates (r, t) .

Tipus d'òptica: ortogonal:

$l, r, b, t, -window-, zNear, zFar$

IDI 2018-2019 1Q

9

The diagram illustrates the OpenGL rendering pipeline. It starts with a camera on a tripod capturing a scene of Albert Einstein. A large blue arrow points from the camera to a close-up view of Einstein's face. Below this, a bracket groups three steps: 1. Posició, orientació (Position, orientation), 2. Òptica (Optics), and 3. Fer la Foto (Take the Photo). The output of step 3 is labeled "Finestra OpenGL (Pantalla)" (OpenGL Window). This window is defined by a red rectangle on a coordinate system with axes xD and yD . The window contains a smaller green rectangle representing the "Viewport/Vista". A mouse cursor is shown clicking on the xD axis of the Viewport. The final output is a windowed image of Einstein, with a green arrow pointing to the label "4. Emmarcar" (Frame).

1. Posició, orientació
2. Òptica } 3. Fer la Foto

Finestra OpenGL (Pantalla)

yD

xD

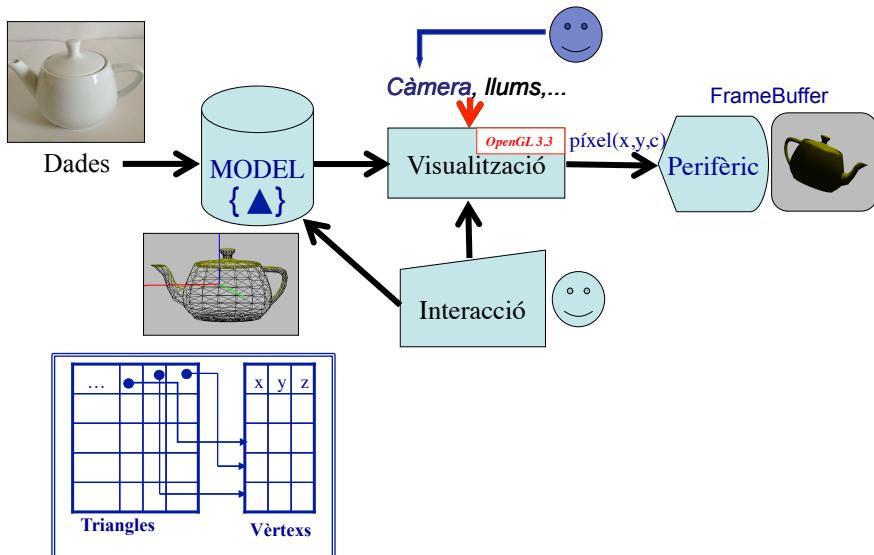
Viewport/Vista

4. Emmarcar

Usualment:

- El viewport és tota la finestra OpenGL
- De moment, no ens preocuparem de si hi ha “deformacions”

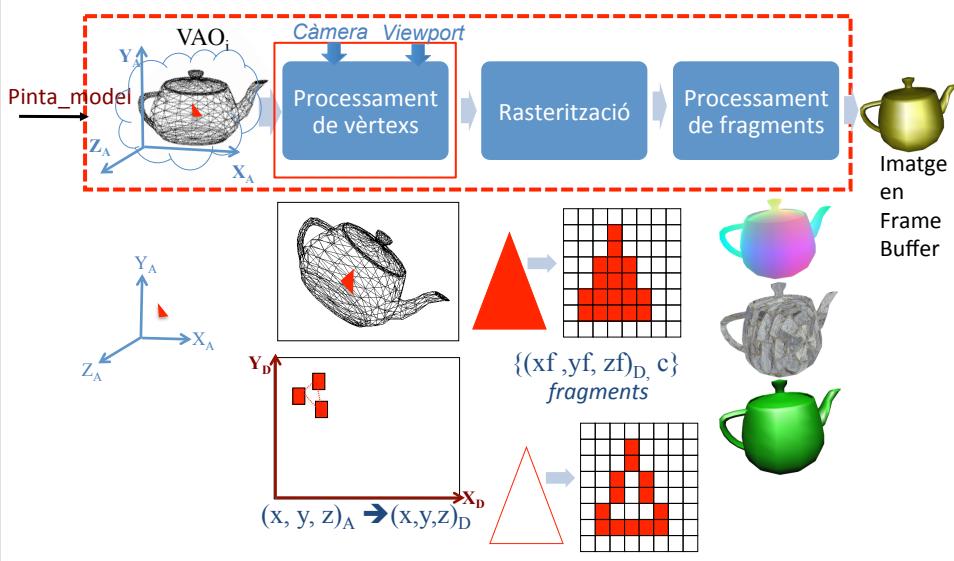
Classe 3: el procés de visualització projectiu



IDI 2018-2019 1Q

11

Paradigma projectiu bàsic



IDI 2018-2019 1Q

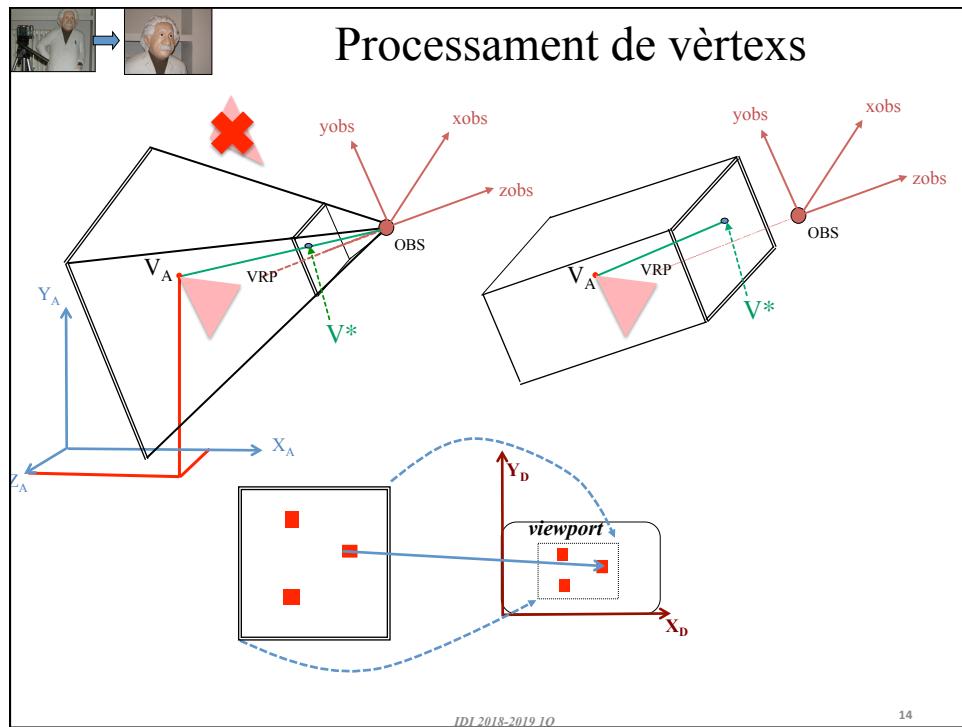
12

Classe 3: contingut

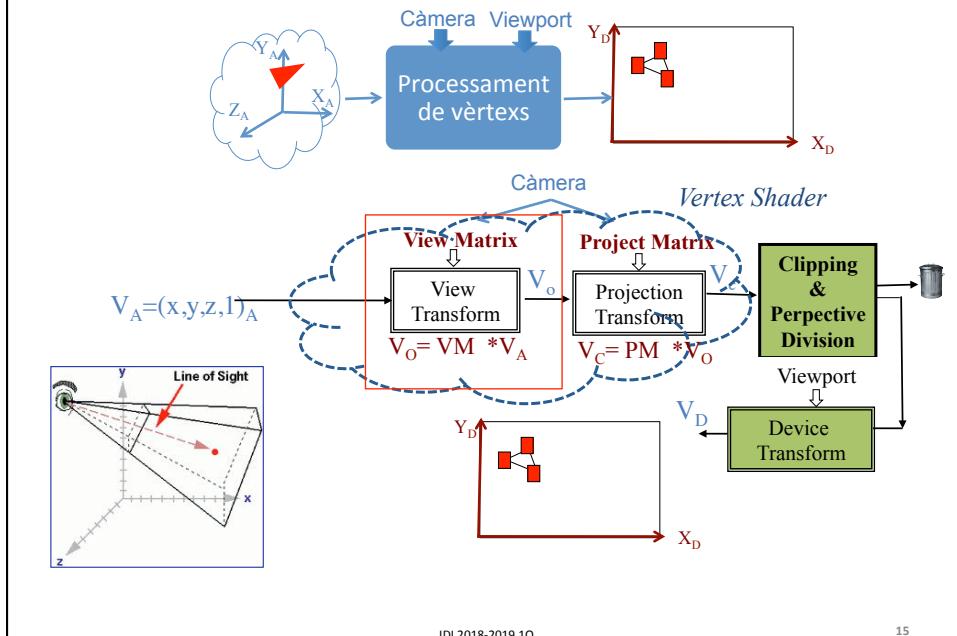
- Especificació de càmera i “viewport”
- El procés de visualització projectiu
- **Processament de vèrtexs:**
 - Seqüència de processos/etapes
 - Matrius requerides
 - **El vertex shader**
- Processament de fragments
 - El fragment shader
- Exemples i primers exercicis

IDI 2018-2019 1Q

13



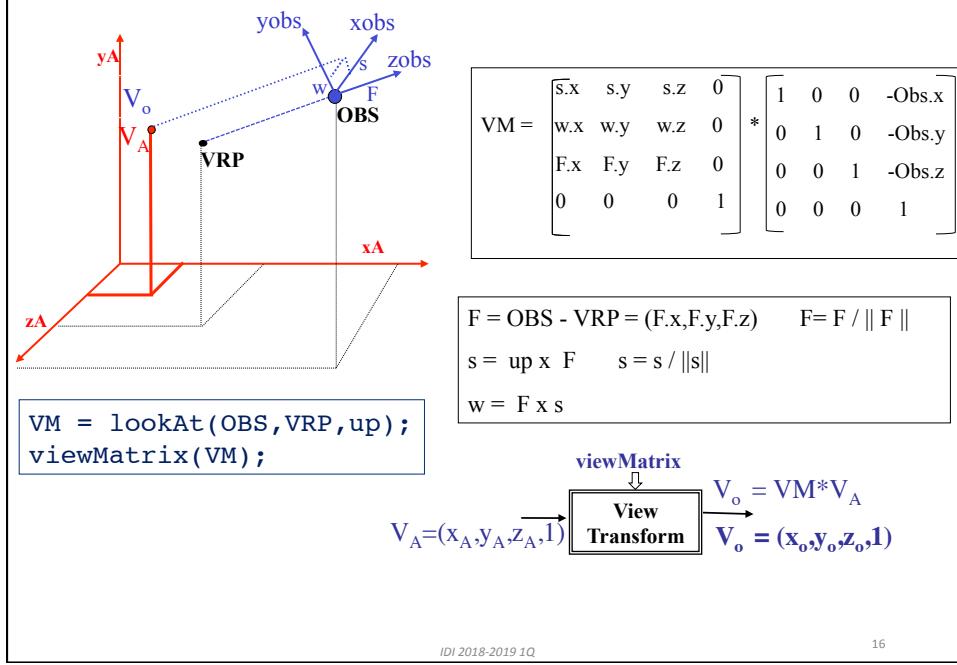
Paradigma projectiu de visualització amb OpenGL 3.3



IDI 2018-2019 1Q

15

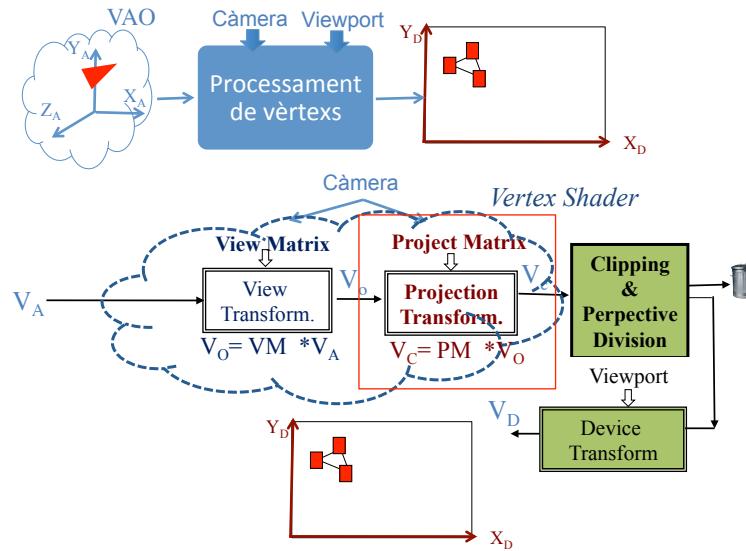
Pas 1: vèrtex en SCO. OBS, VRP, up → Càlcul de la viewMatrix



IDI 2018-2019 1Q

16

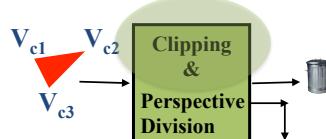
Pas 2: vèrtex en Coord Clipping. Òptica → Càlcul de la projectMatrix



IDI 2018-2019 1Q

17

Per què transformar el punt a coordenades de clipping simplifica el clipping?



$$V_c = (x_c, y_c, z_c, w_c)$$

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

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

IDI 2018-2019 1Q

18

Pas 2: Obtenir punts en Coord. Clipping

$V_o = (x_o, y_o, z_o, 1)_o \xrightarrow{\text{ProjectMatrix}} V_c = PM * V_o$

$PM = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $a = 2/(r-l)$ $b = 2/(t-b)$
 $c = 2/(zf-zn)$ $d = (zn+zf)/(zf-zn)$

Clipping & Perspective Division

$V_c = (x_c, y_c, z_c, w_c)$ on $w_c = 1$

$-1 \leq x_c \leq 1$
 $-1 \leq y_c \leq 1$
 $-1 \leq z_c \leq 1$

Optica Ortogonal

IDI 2018-2019 IQ

19

Pas 2: Obtenir punts en Coord. Clipping

$V_o = (x_o, y_o, z_o, 1)_o \xrightarrow{\text{ProjectMatrix}} V_c = PM * V_o$

$PM = \begin{pmatrix} 1/r_a * a & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix}$ $a = \tan(\text{FOV}/2)$
 $c = (zf+zn)/(zn-zf)$
 $d = 2*zn*zf/(zn-zf)$

Clipping & Perspective Division

Optica Perspectiva

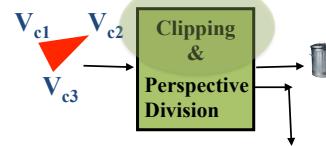
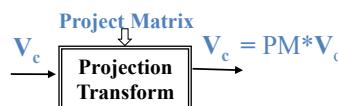
$V_c = (x_c, y_c, z_c, w_c)$ on $w_c = -z_o$

$-w_c \leq x_c \leq w_c$
 $-w_c \leq y_c \leq w_c$
 $-w_c \leq z_c \leq w_c$

IDI 2018-2019 IQ

20

Pas 2: Obtenir punts en Coord. Clipping



$$PM = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{aligned} a &= 2/(r-l) & b &= 2/(t-b) \\ c &= 2/(f-n) & d &= (n+f)/(f-n) \end{aligned}$$

```
PM=ortho(l,r,b,t,zN,ZF);
projectMatrix(PM);
```

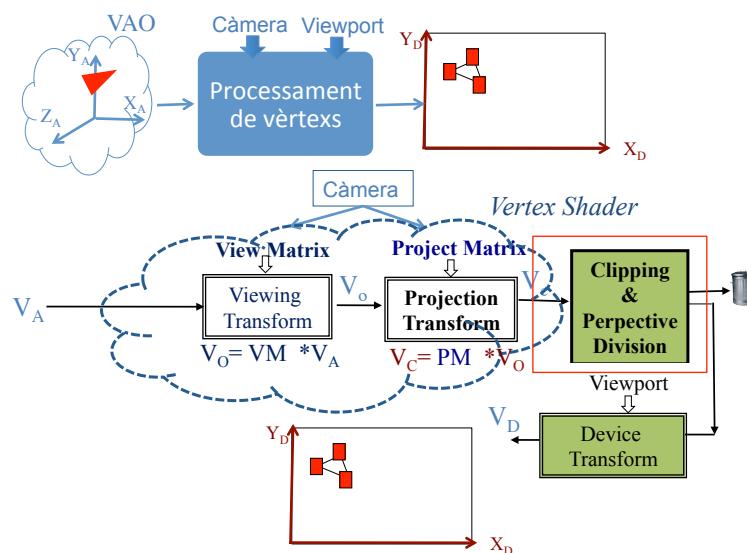
$$PM = \begin{pmatrix} 1/r*a & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad \begin{aligned} a &= \tan(\text{FOV}/2) \\ c &= (f+n)/(n-f) \\ d &= 2nf/(n-f) \end{aligned}$$

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

IDI 2018-2019 1Q

21

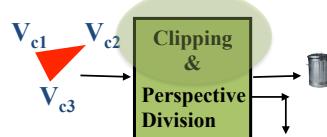
Pas 3: Clipping i projecció



IDI 2018-2019 1Q

22

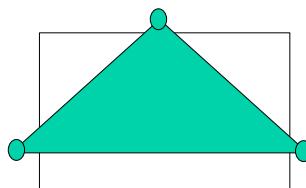
Per què transformar el punt a coordenades de clipping simplifica el clipping?



$$V_c = (x_c, y_c, z_c, w_c)$$

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

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

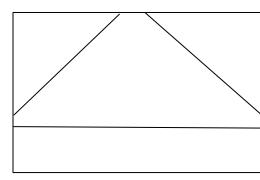
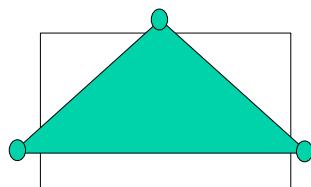


IDI 2018-2019 1Q

23

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

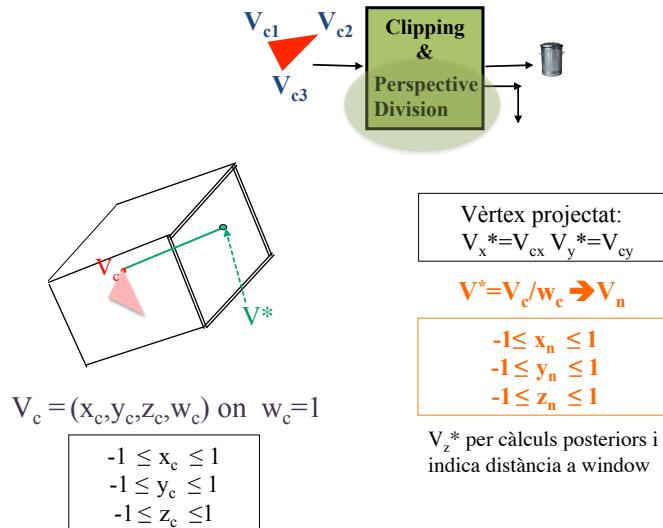
$$\begin{aligned} -w_c \leq x_c \leq w_c \\ -w_c \leq y_c \leq w_c \\ -w_c \leq z_c \leq w_c \end{aligned}$$



IDI 2018-2019 1Q

24

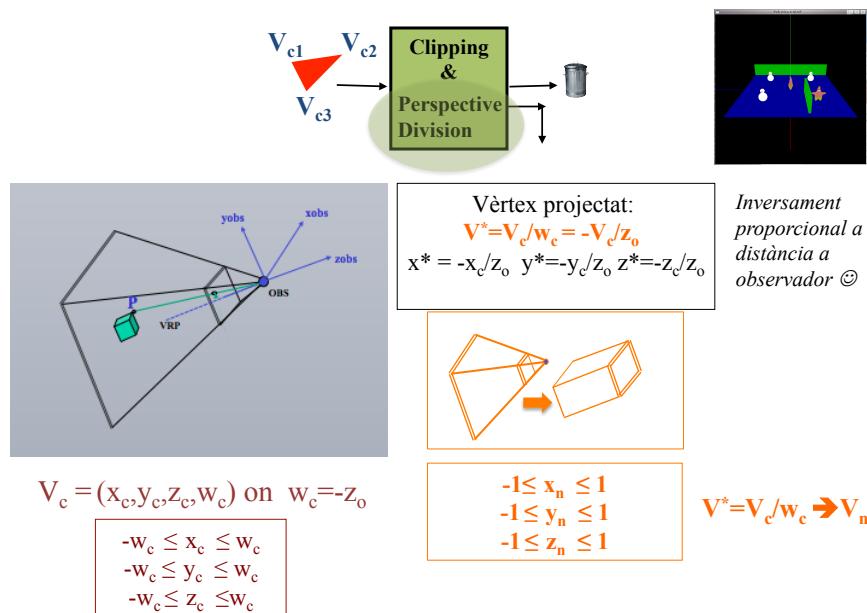
Per què les matrius són diferents? → Càlcul punt projectat



IDI 2018-2019 1Q

25

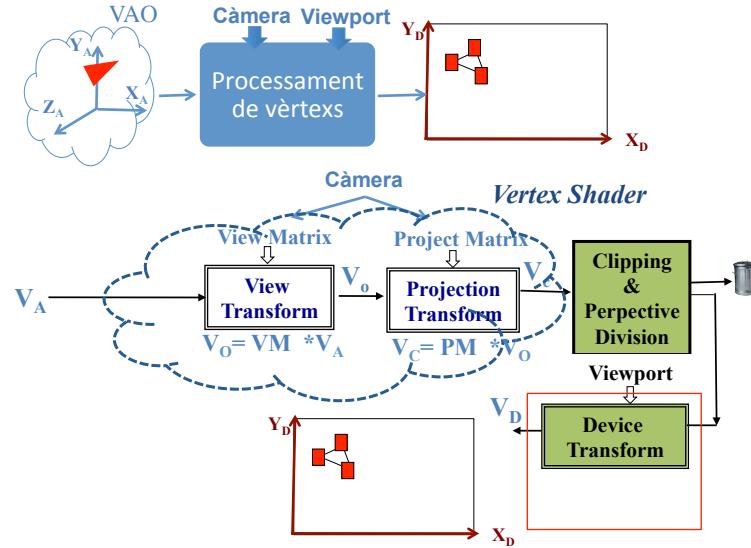
Per què les matrius són diferents? → Càlcul punt projectat



IDI 2018-2019 1Q

26

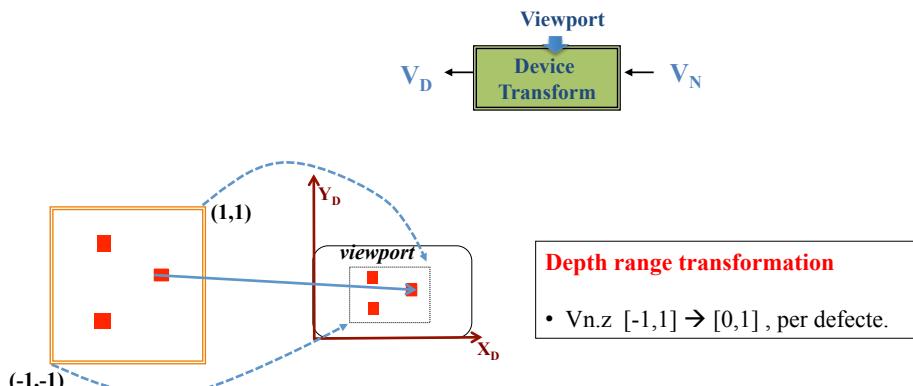
Pas 4: Transformació a coordenades de dispositiu



IDI 2018-2019 1Q

27

Pas 4: Transformació a coordenades de dispositiu



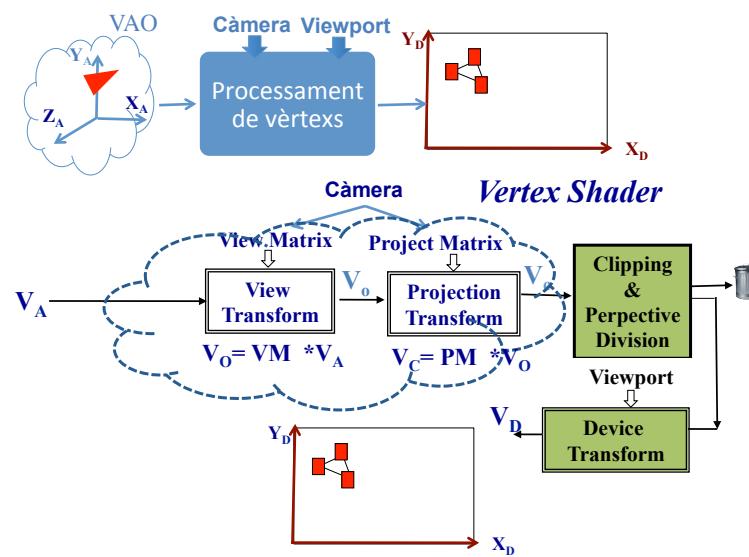
IDI 2018-2019 1Q

28

Classe 3: contingut

- Especificació de càmera i “viewport”
- El procés de visualització projectiu
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - **El vertex shader**
- Processament de fragments
 - El fragment shader
- Exemples i primers exercicis

Processat de vèrtex: Què cal que programem?



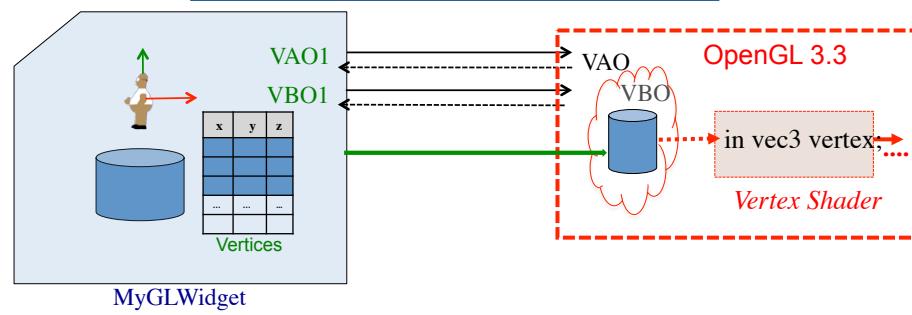
Processat de vèrtex: Què cal que programem?

```
#version 330 core
```

```
in vec3 vertex;
uniform mat4 PM;
uniform mat4 VM;
```

```
void main() {
    gl_Position = PM*VM*vec4 (vertex, 1.0);
}
```

Vertex Shader



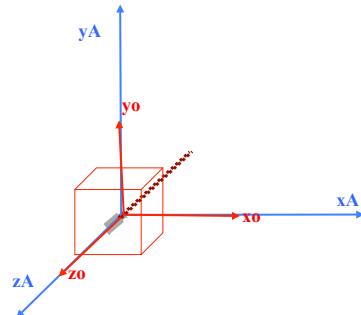
IDI 2018-2019 1Q

32

Processat de vèrtex: Què cal que programem?

Vertex Shader

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



Volum de Visió cub de (-1,-1,-1) a (1,1,1)

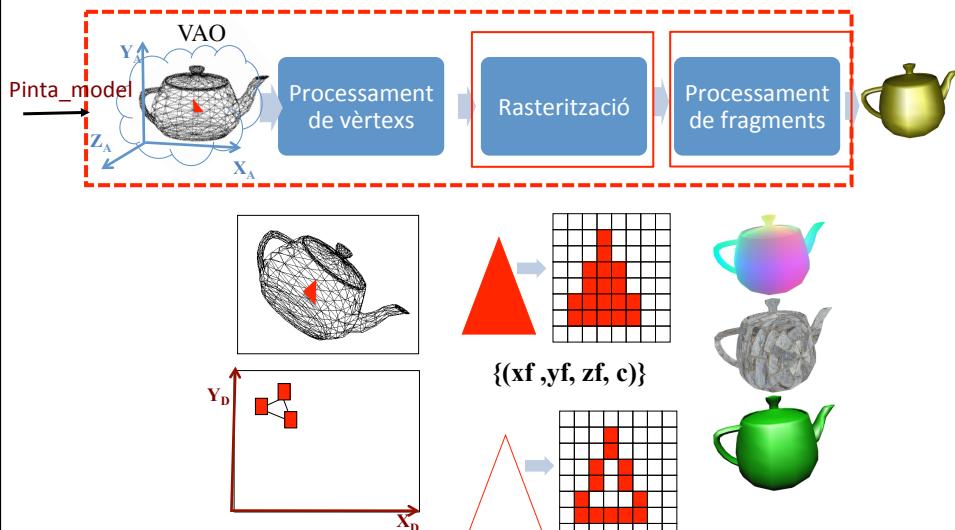
IDI 2018-2019 1Q

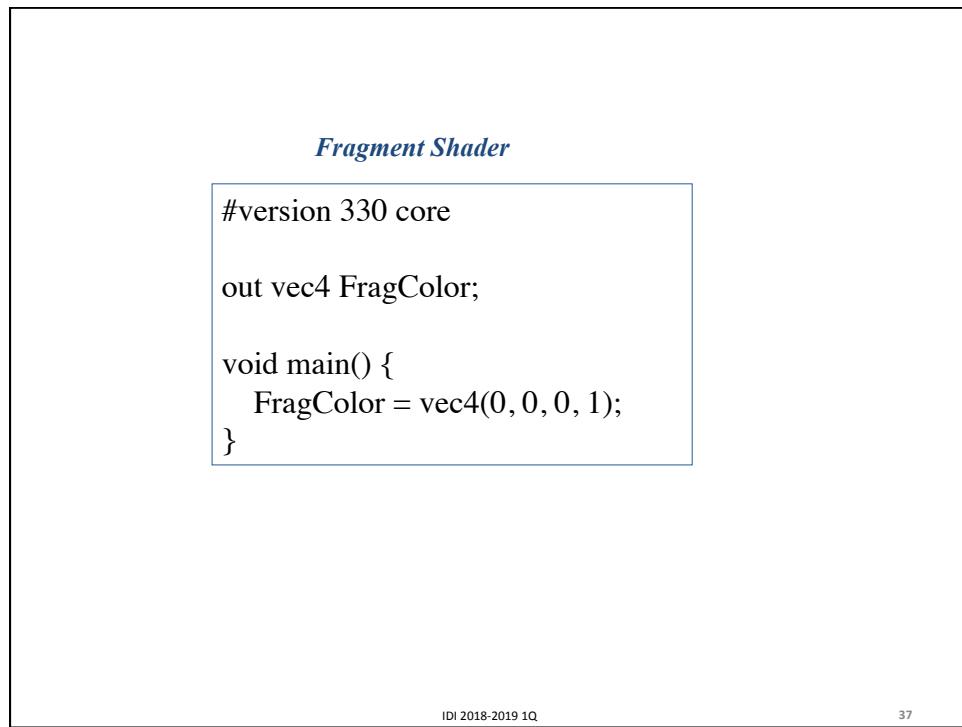
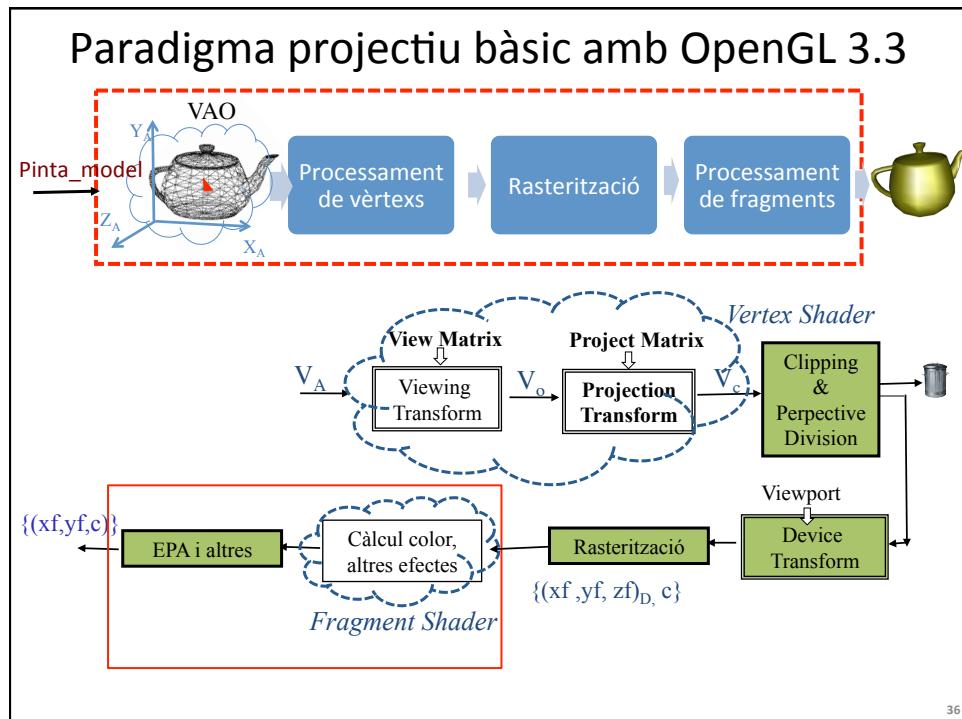
33

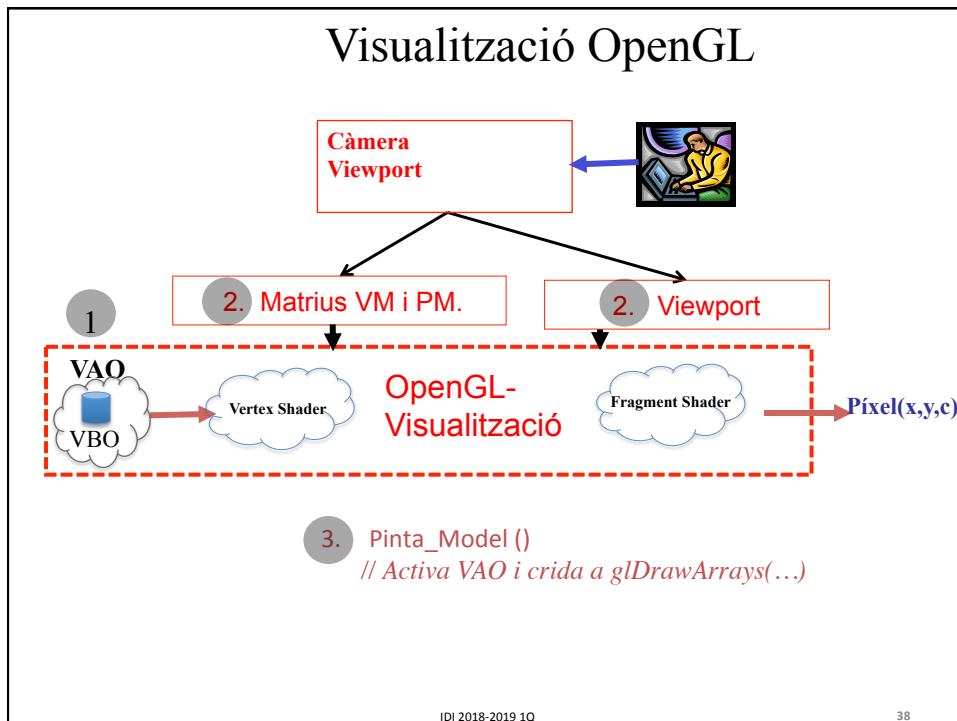
Classe 3: contingut

- Especificació de càmera i “viewport”
- El procés de visualització projectiu
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- **Processament de fragments**
 - **El fragment shader**
- Exemples i primers exercicis

Paradigma projectiu bàsic amb OpenGL 3.3





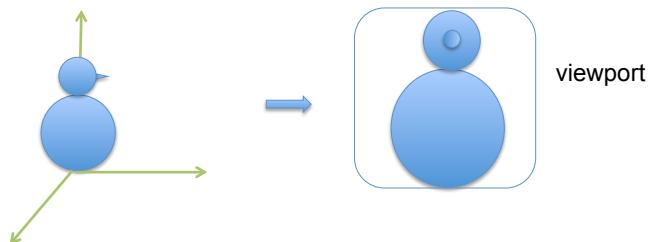


Classe 3: contingut

- Especificació de càmera i “viewport”
- El procés de visualització projectiu
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments
 - El fragment shader
- **Exemples i primers exercicis**

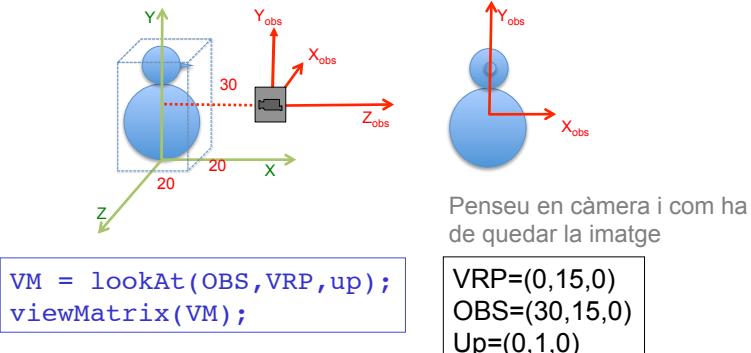
Exemple 1: Donada una funció `pinta_ninot()` que pintaria un objecte com el de la figura, format per: una esfera de radi 10 i centre (0,10,0), una altra esfera de radi 5 i centre (0,25,0), i un con de base centrada en (2.5, 25,0), r=2 i llargada 5 orientat segons l'eix X⁺

- Indica 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 2018-2019 1Q

40



IDI 2018-2019 1Q

41

1. Posició, orientació
2. Òptica perspectiva

Viewport: tota la finestra gràfica 600x600
- Ninot optimitzi espai en viewport
- Sense deformacions

IDI 2018-2019 1Q

42

Exemple 1: Òptica perspectiva

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

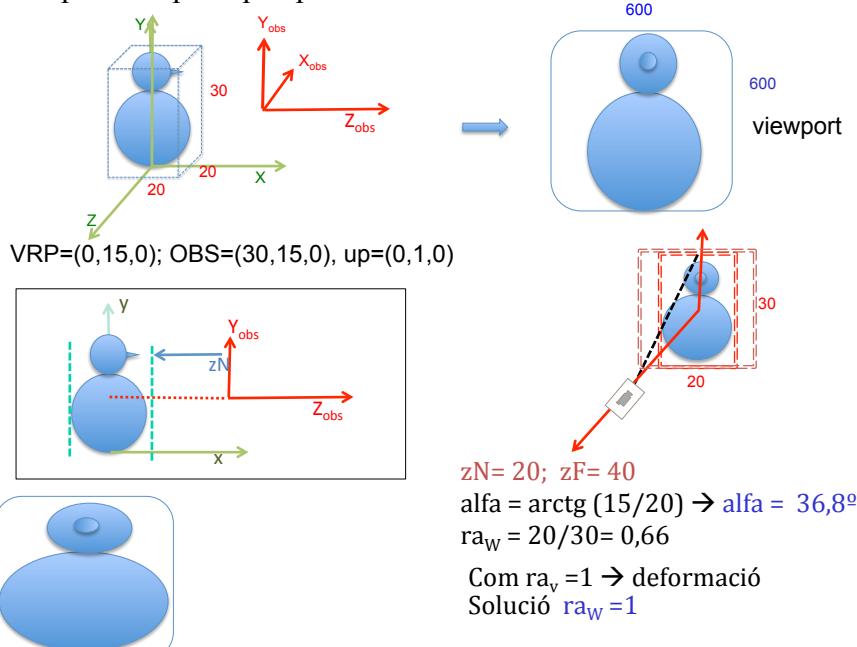
$\text{alfa} = \arctg (15/20) \rightarrow \text{alfa} = 36,8^\circ$
 $r_{aW} = 20/30 = 0,66$

Com $r_{aV}=1 \rightarrow$ deformació
Solució $r_{aW}=1$

IDI 2018-2019 1Q

43

Exemple 1: Òptica perspectiva

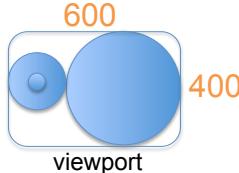


IDI 2018-2019 1Q

44

Per pensar...

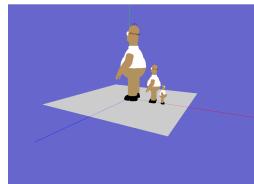
- Com fer un zoom? Quins paràmetres de càmera modificaries?
- Quins paràmetres de posicionament de càmera per a obtenir:



- Quins paràmetres d'òptica ortogonal?
- Quins paràmetres amb òptica perspectiva però viewport 600x600?

IDI 2018-2019 1Q

45



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?