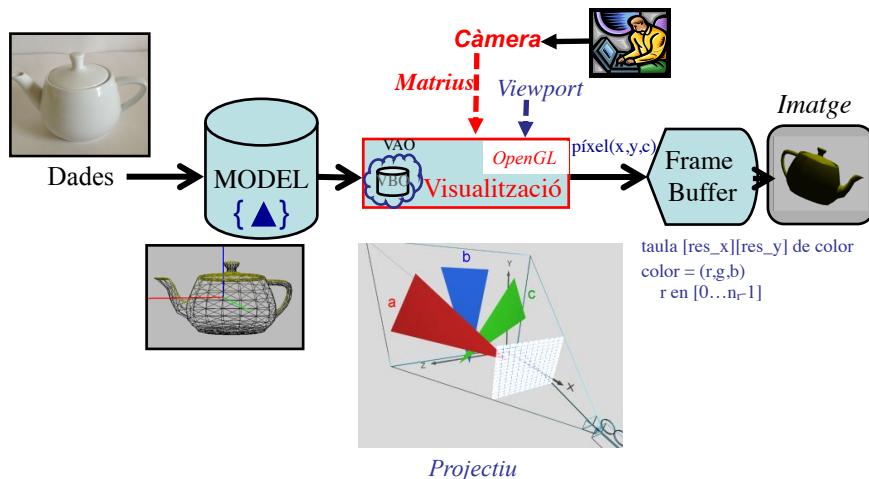


Contingut



IDI Q2 2018-2019

Classe 3: Contingut

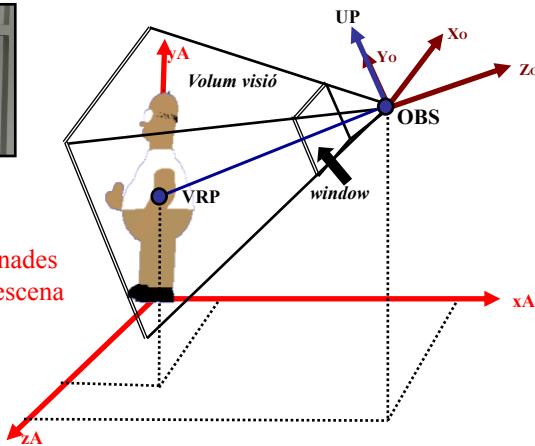
- Especificació de càmera
- Exemple
- El procés de visualització projectiu (breu repàs)
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments:
 - El fragment shader

IDI Q2 2018-2019

Com especificar la càmera virtual?



Sistema Coordenades
Aplicació/món/escena

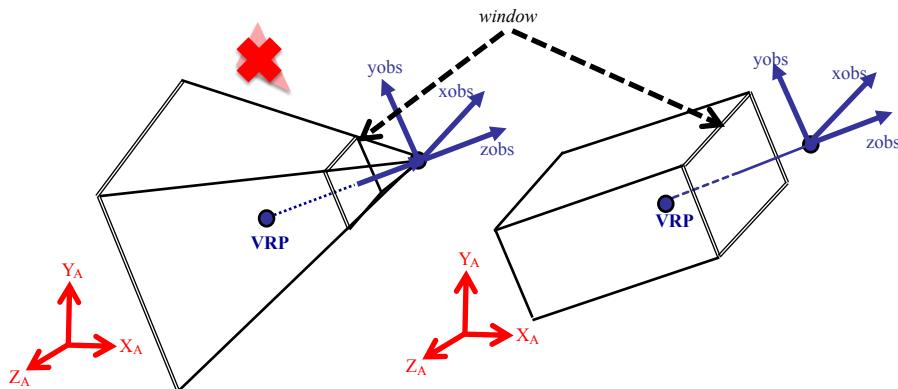


1. Ubicació càmera respecte SCA: obs, vrp, up
2. Definir òptica: Volum de Visió → window, zNear, zFar

IDI Q2 2018-2019

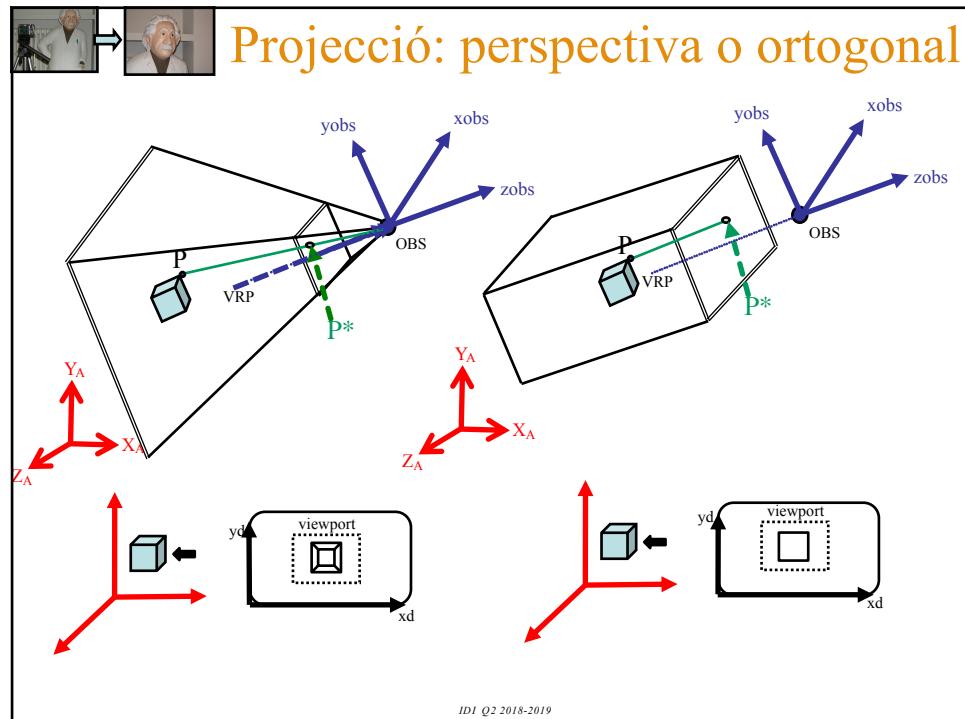


Òptica: perspectiva o ortogonal

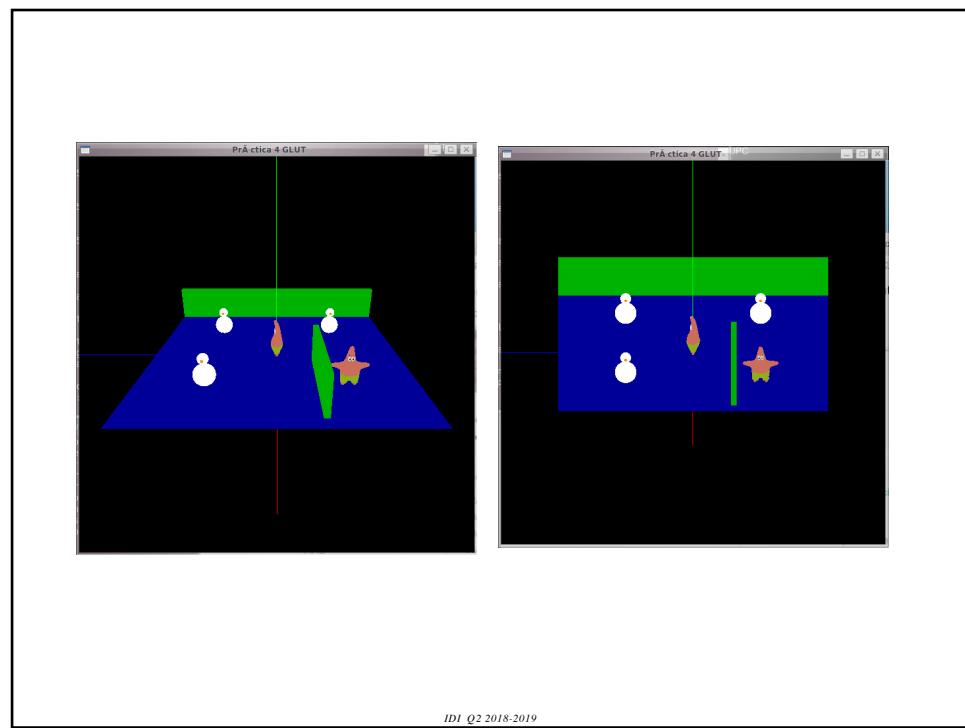


Definir òptica: volum de visió → (window, zNear, zFar)

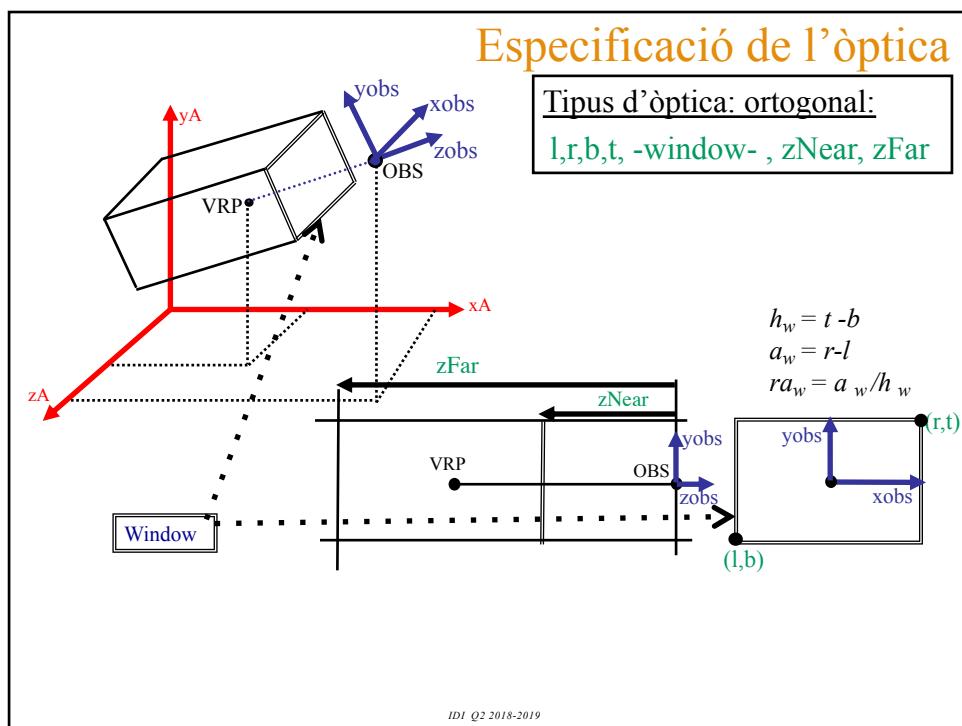
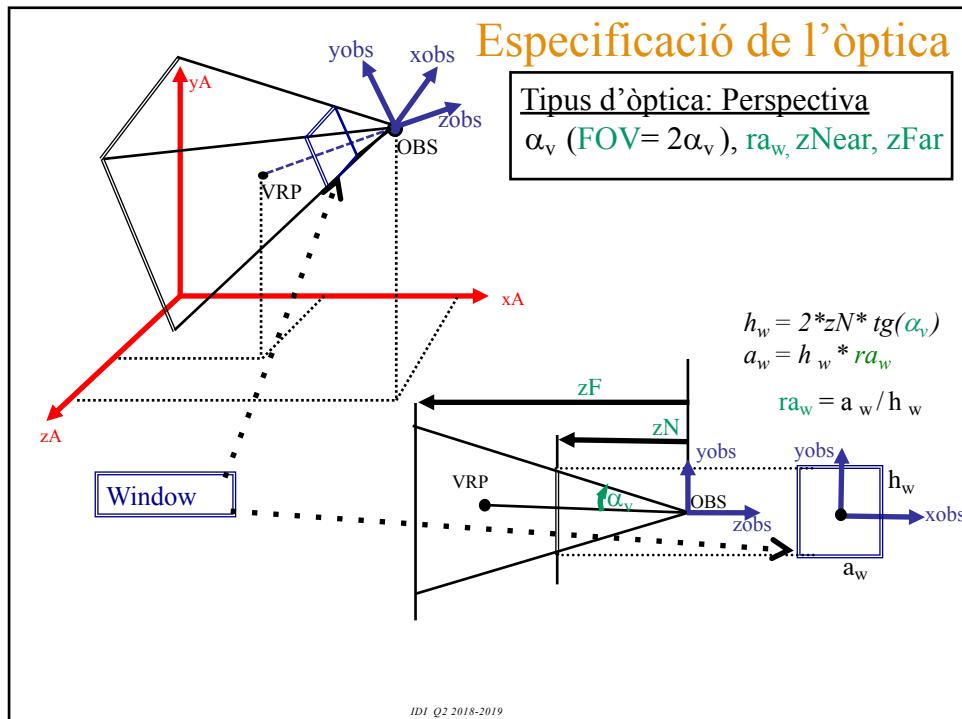
IDI Q2 2018-2019



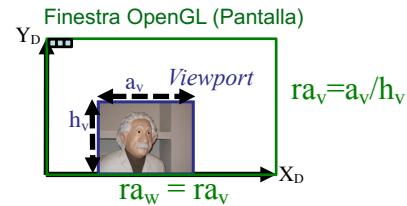
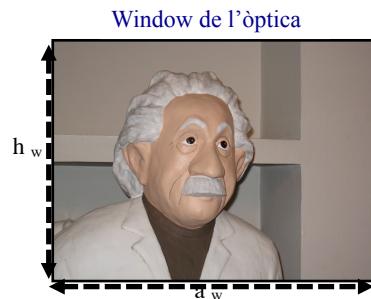
IDI Q2 2018-2019



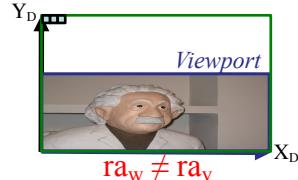
IDI Q2 2018-2019



Sobre la relació d'aspecte del window i del viewport



`glViewport (ox, oy, av, hv)`



Per a no tenir deformació en la imatge

$$ra_w = ra_v$$

IDI Q2 2018-2019

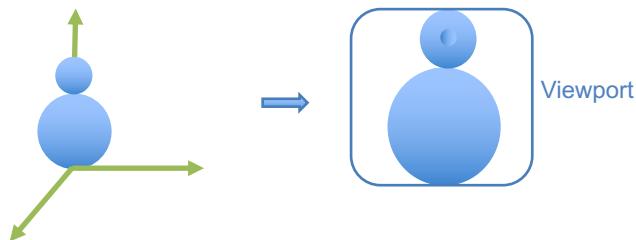
Classe 3: Contingut

- Especificació de càmera
- **Exemple**
- El procés de visualització projectiu (breu repàs)
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments:
 - El fragment shader

IDI Q2 2018-2019

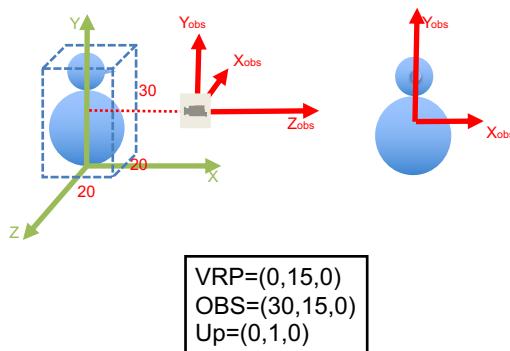
Exemple 1: Donada una funció `pinta_ninot()` que envia a visualitzar el VAO d'un objecte com el de la figura, format per: una esfera de radi 10 i centre $(0,10,0)$, una altra esfera de radi 5 i centre $(0,25,0)$, i un con de base centrada en $(2.5, 25, 0)$, $r=2$ i llargada 5 orientat segons l'eix X^+

Indica tots els paràmetres d'una càmera que permeti obtenir la imatge similar a la mostrada en l'esquema de la Figura. El viewport de 600×600 ocupa tota la finestra gràfica.



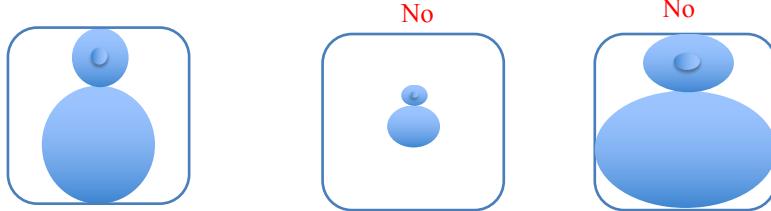
IDI Q2 2018-2019

Exemple 1. Posició, orientació de la càmera



IDI Q2 2018-2019

Exemple 1. Òptica de la càmera

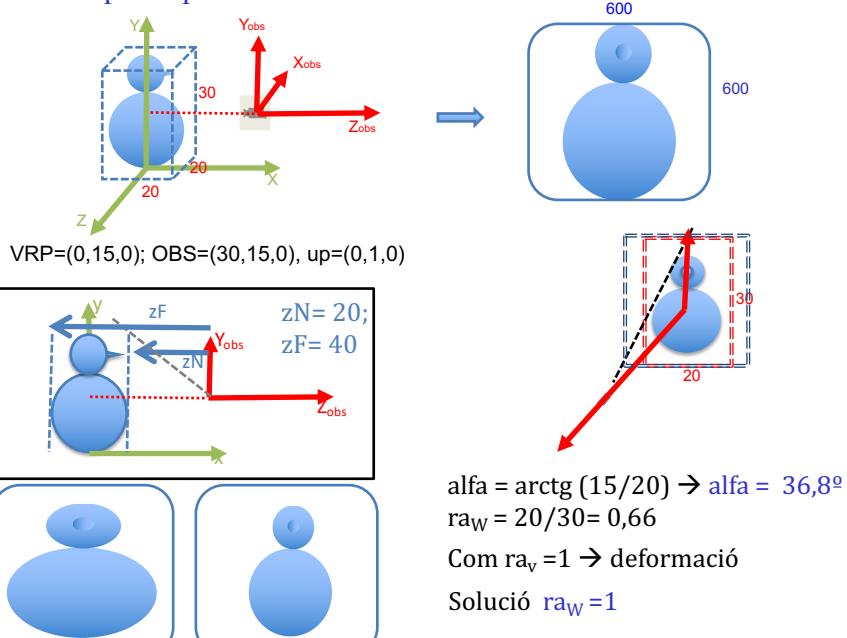


Restriccions:

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

IDI Q2 2018-2019

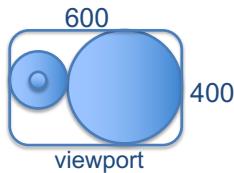
Exemple. Òptica de la càmera



IDI Q2 2018-2019

Per pensar...

- Quins paràmetres requiria una òptica ortogonal? Quin efecte tindria en la imatge generada?
- Quins paràmetres de posicionament de càmera per a obtenir:



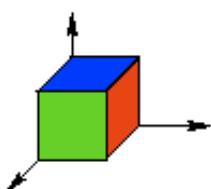
- Quins paràmetres per una òptica perspectiva amb un de viewport 600x600?
- Com fer un zoom? Quins paràmetres de la càmera modificaries?

IDI Q2 2018-2019

Exemple 2. Tenim una escena amb un cub de costat 2 orientat amb els eixos i de manera que el seu vèrtex mínim està situat a l'origen de coordenades. La cara del cub que queda sobre el pla $x=2$ és de color vermell, la cara que queda sobre el pla $z=2$ és de color verd i la resta de cares són blaves.

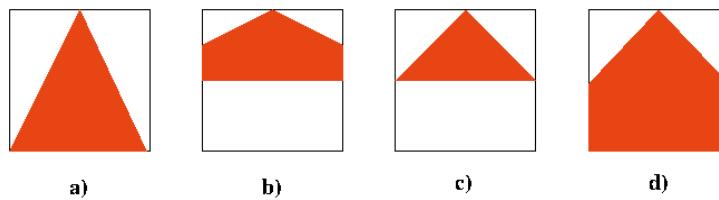
a) Indica TOTS els paràmetres d'una càmera perspectiva que permeti veure completes a la vista només les cares vermella i verda. La relació d'aspecte del viewport (vista) és 2. Fes un dibuix indicant la imatge final que s'obtindria.

b) Quin efecte tindria en la imatge final modificar l'òptica ortogonal?
Defineix la càmera ortogonal.



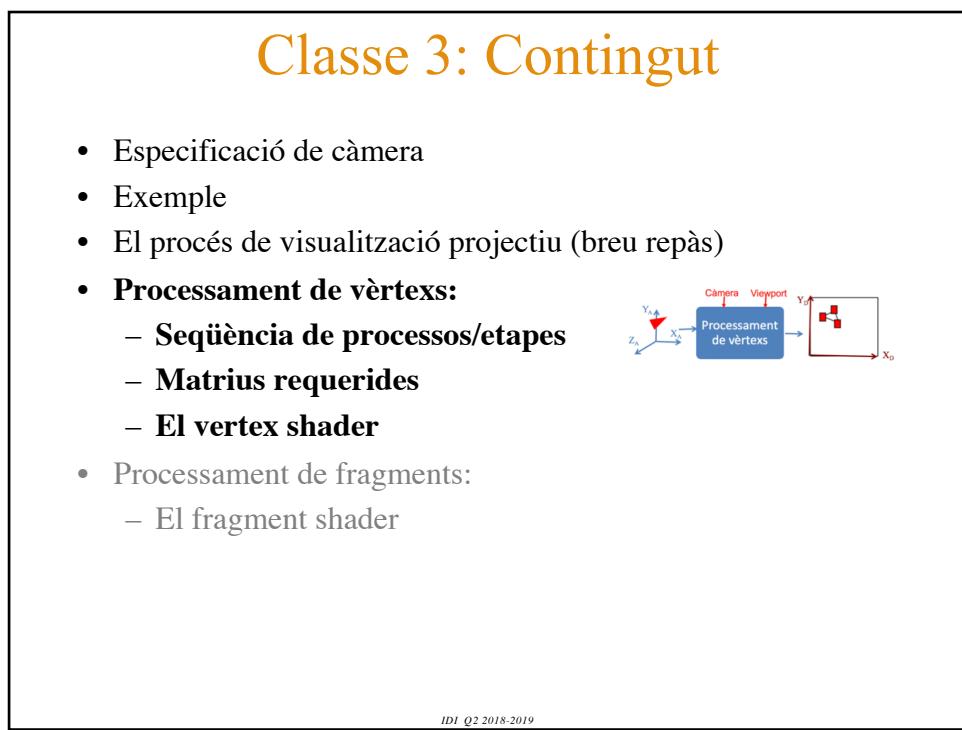
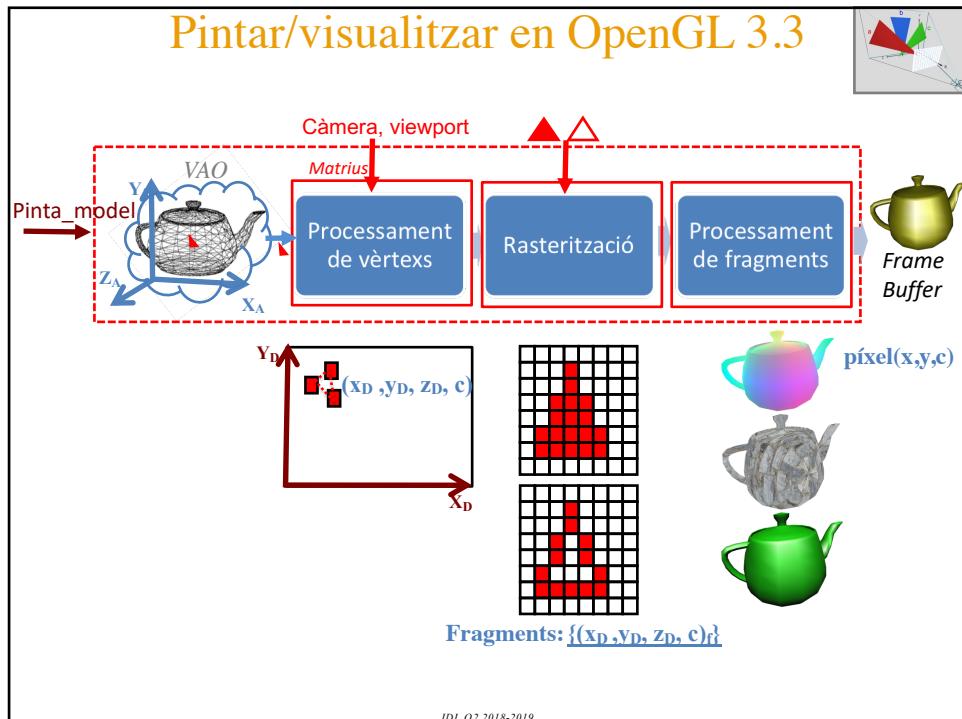
Exemple 3

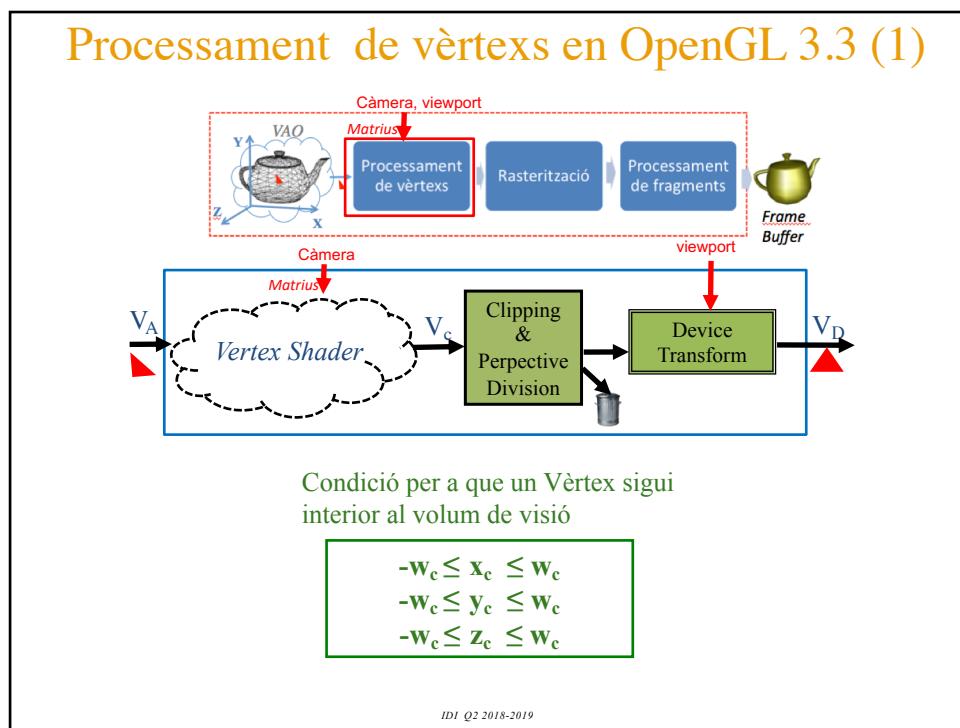
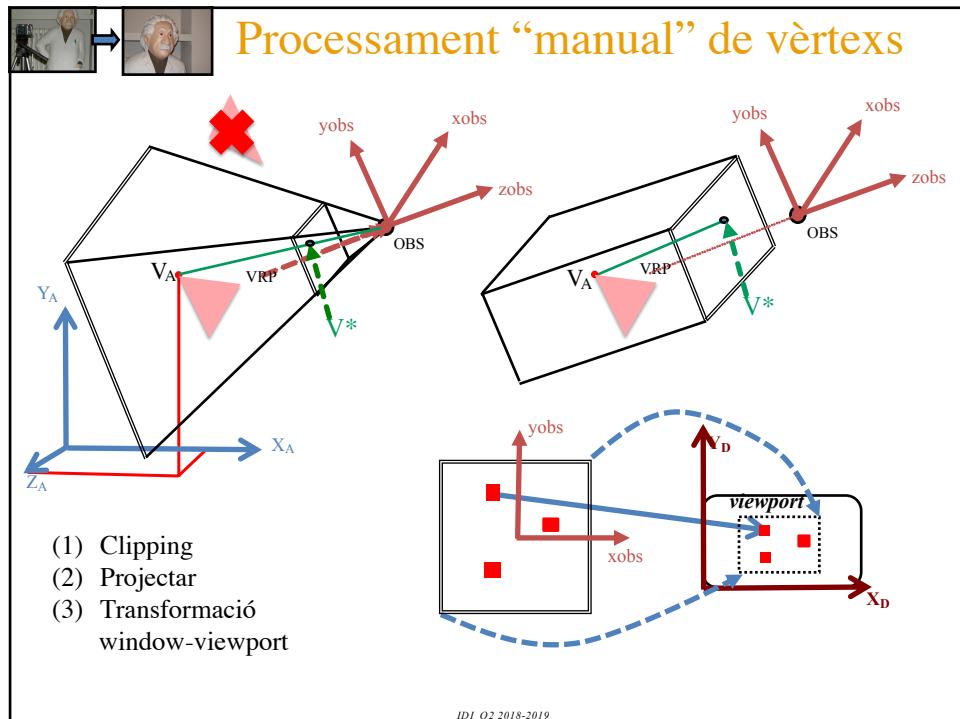
Tenim una escena amb un triangle vermell amb vèrtexs $V1=(-2,0,0)$, $V2 = (2, 0, 0)$ i $V3= (0, 1, 0)$. Suposant que tenim un viewport quadrat de 600x600 pixels, i que hem inicialitzat les matrius de càmera (view) i projecció (proj) a la matriu identitat, indica quina de les següents imatges és la que sortirà en un viewport de 600x600 (sabem que el Vertex Shader i el Fragment Shader estan correctament implementats):



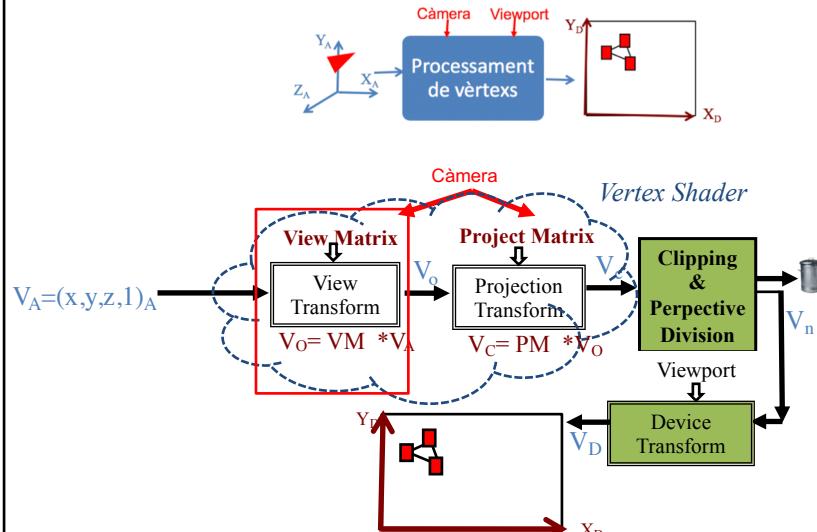
Classe 3: Contingut

- Especificació de càmera
- Exemple
- **El procés de visualització projectiu (breu repàs)**
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments:
 - El fragment shader



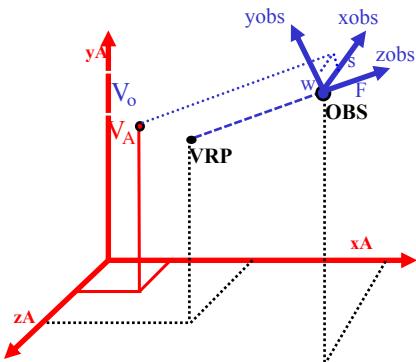


Processament de vèrtexs en OpenGL 3.3 (2)



IDI Q2 2018-2019

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



$$VM = \begin{bmatrix} s.x & s.y & s.z & 0 \\ w.x & w.y & w.z & 0 \\ F.x & F.y & F.z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & -Obs.x \\ 0 & 1 & 0 & -Obs.y \\ 0 & 0 & 1 & -Obs.z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F = OBS - VRP = (F.x, F.y, F.z) \quad F = F / \| F \|$$

$$s = up \times F \quad s = s / \| s \|$$

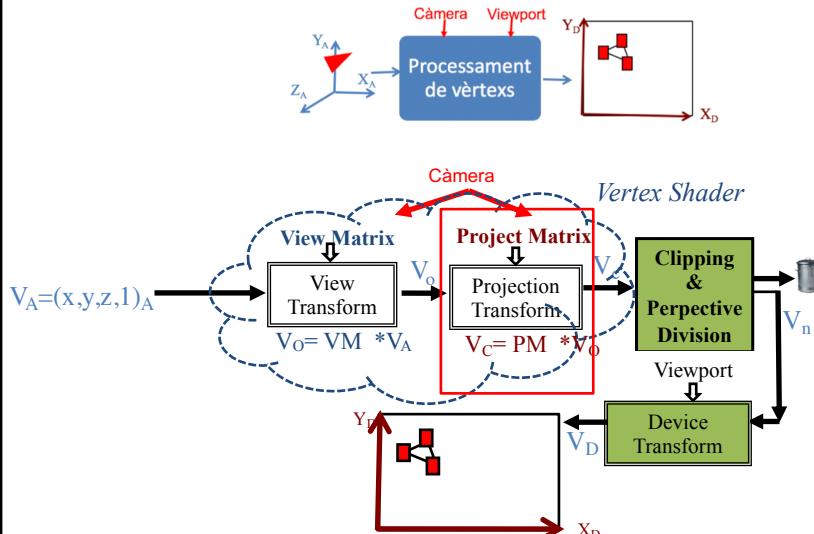
$$w = F \times s$$

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

$$V_A = (x_A, y_A, z_A, 1) \xrightarrow{\text{viewMatrix}} V_o = VM * V_A \xrightarrow{\text{View Transform}} V_o = (x_o, y_o, z_o, 1)$$

IDI Q2 2018-2019

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



IDI Q2 2018-2019

Pas 2: La projectMatrix

$$V_o = (x, y, z, 1)_o \xrightarrow{\text{ProjectMatrix}} V_c = PM * V_o \xrightarrow{\text{Projection Transform.}}$$

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

Òptica Ortogonal

$V_c = (x_c, y_c, z_c, w_c)$ on $w_c = 1$
 $PM = \text{ortho}(l, r, b, t, zN, ZF);$
 $\text{projectMatrix}(PM);$

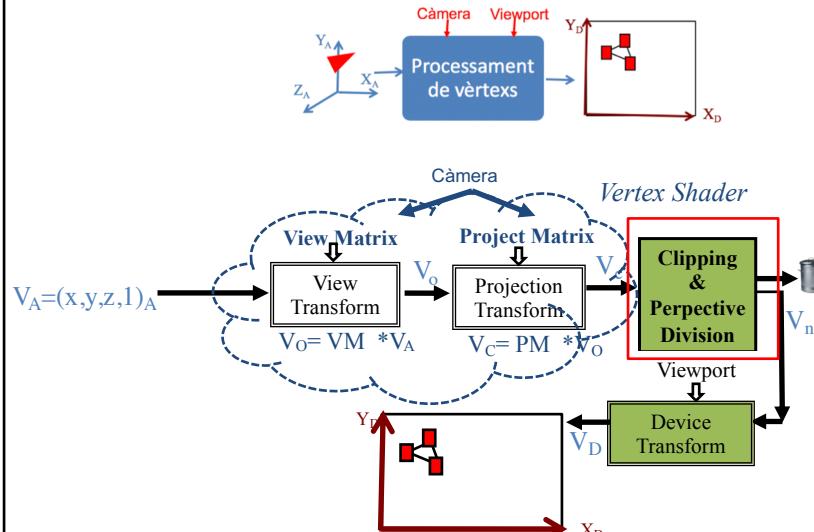
$$PM = \begin{pmatrix} 1/r * a & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad \begin{aligned} a &= \tan(\text{FOV}/2) \\ c &= (zf+zn)/(zn-zf) \\ d &= 2*zn * zf / (zn-zf) \end{aligned}$$

Òptica Perspectiva

$V_c = (x_c, y_c, z_c, w_c)$ on $w_c = -z_o$
 $PM = \text{perspective}(\text{FOV}, ra, zN, ZF);$
 $\text{projectMatrix}(PM);$

IDI Q2 2018-2019

Pas 3: Clipping i projecció



IDI Q2 2018-2019

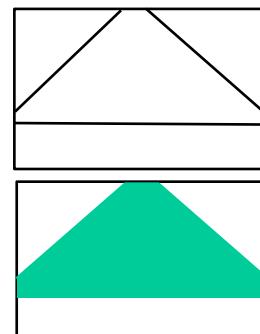
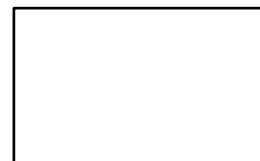
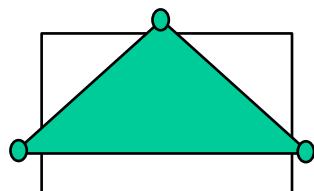
Pas 3.1 : Clipping



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

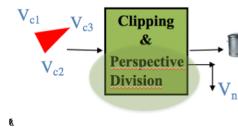
$$\begin{aligned} -w_c \leq x_c \leq w_c \\ -w_c \leq y_c \leq w_c \\ -w_c \leq z_c \leq w_c \end{aligned}$$

$V_c = (x_c, y_c, z_c, w_c)$ on $w_c=1$ en ortogonal
 $V_c = (x_c, y_c, z_c, w_c)$ on $w_c=-z_0$ en perspectiva



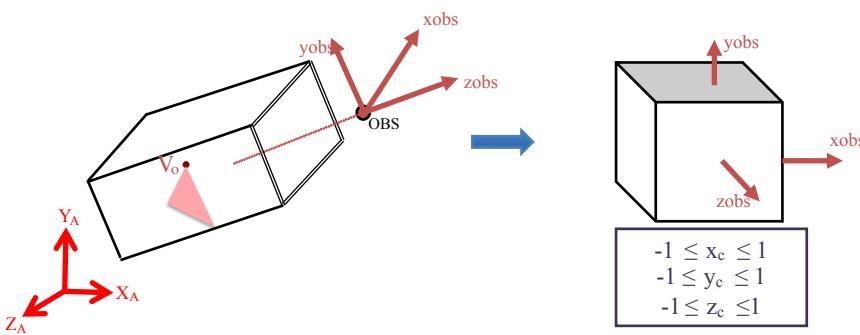
IDI Q2 2018-2019

Pas 3.2 : Projcció. Òptica ortogonal (1)



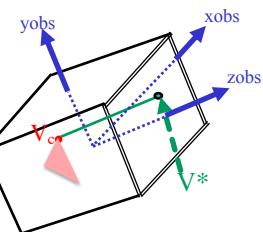
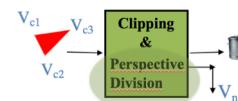
$$\text{PM} = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{aligned} a &= 2/(r-l) & b &= 2/(t-b) \\ c &= 2/(zf-zn) & d &= (zn+zf)/(zf-zn) \end{aligned}$$

$$V_c = (x_c, y_c, z_c, w_c) \text{ on } w_c = 1$$



IDI Q2 2018-2019

Pas 3.2 : Projcció. Òptica ortogonal (2)



$$\text{Vèrtex projectat: } V_x^* = V_{cx}, V_y^* = V_{cy}$$

$$V^* = V_c / w_c \rightarrow V_n$$

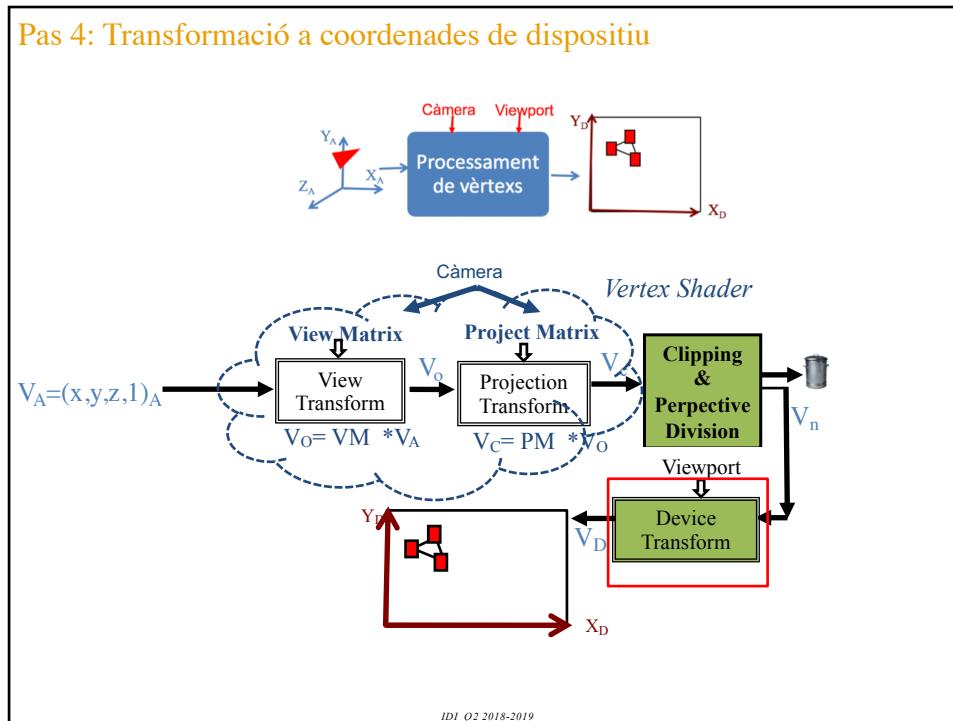
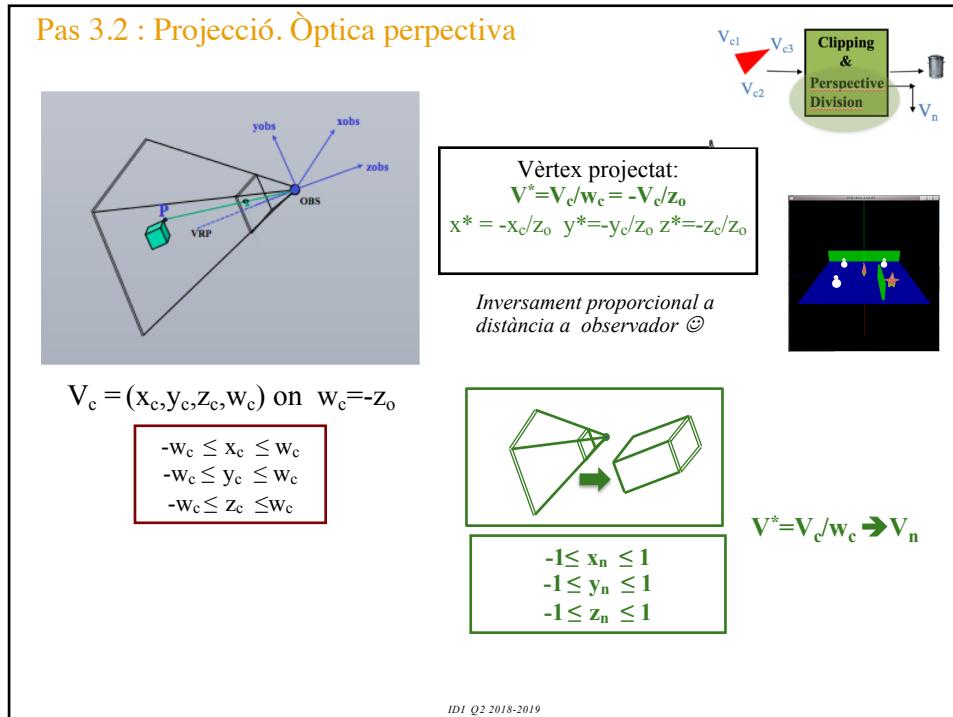
$$\begin{aligned} -1 \leq x_n &\leq 1 \\ -1 \leq y_n &\leq 1 \\ -1 \leq z_n &\leq 1 \end{aligned}$$

$$V_c = (x_c, y_c, z_c, w_c) \text{ on } w_c = 1$$

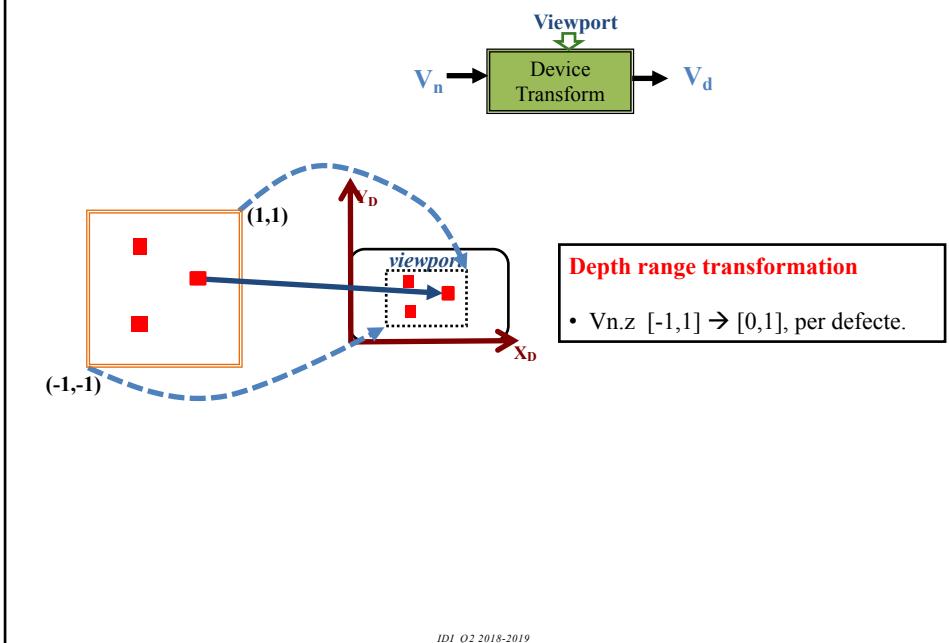
$$\begin{aligned} -1 \leq x_c &\leq 1 \\ -1 \leq y_c &\leq 1 \\ -1 \leq z_c &\leq 1 \end{aligned}$$

V_z^* per càlculs posteriors i
indica distància a window

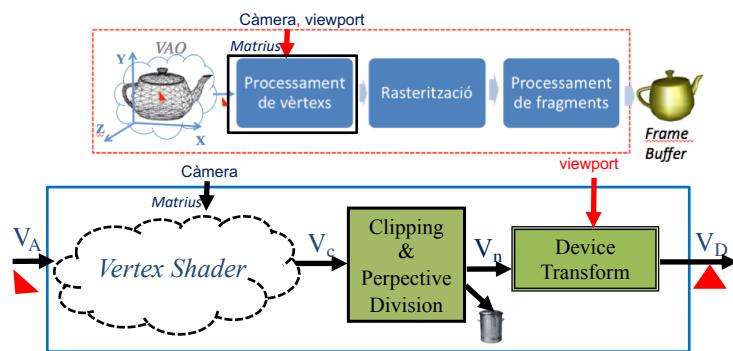
IDI Q2 2018-2019



Pas 4: Transformació a coordenades de dispositiu



Processament de vèrtexs en OpenGL 3.3 (3)



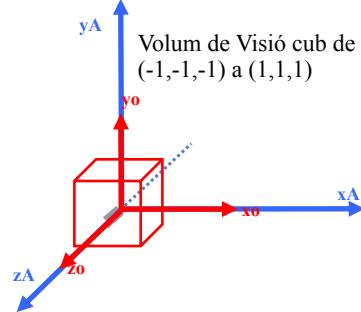
Processat de vèrtex: El vèrtex Shader

Vertex Shader

```
#version 330 core
in vec3 vertex;
uniform mat4 PM;
uniform mat4 VM;

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

```
#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);
}
```

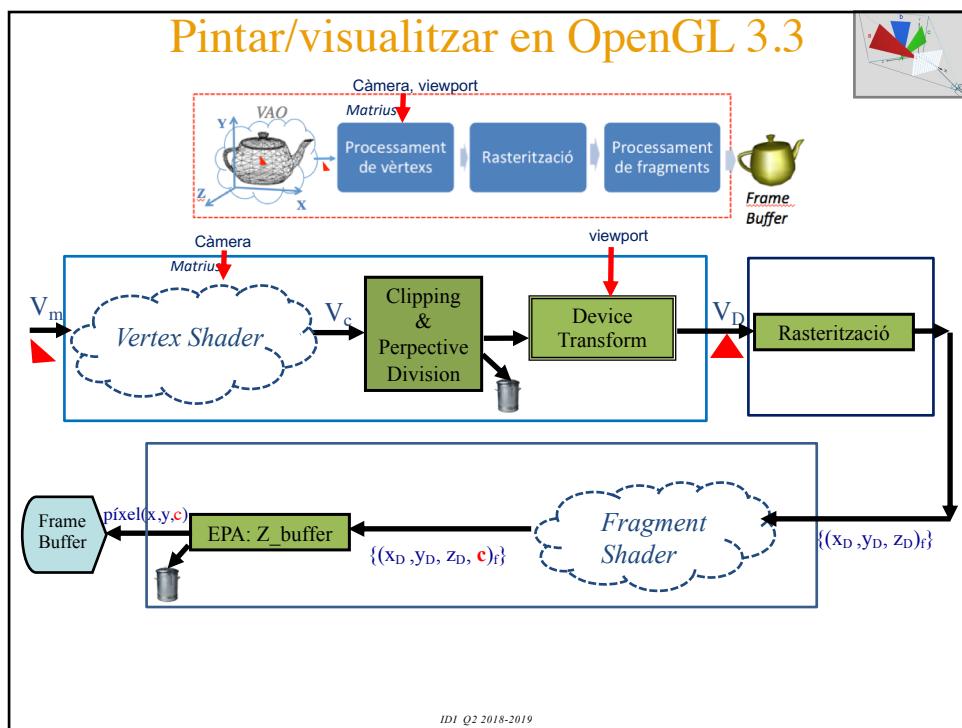
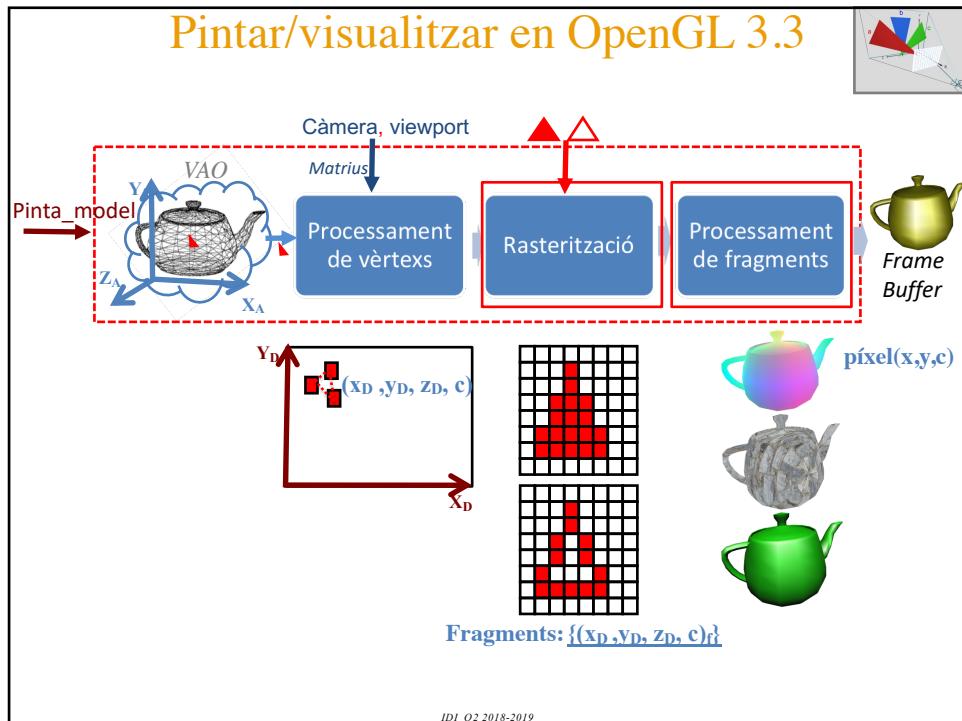


IDI_Q2 2018-2019

Classe 3: Contingut

- Especificació de càmera
- Exemple
- El procés de visualització projectiu (breu repàs)
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- **Processament de fragments:**
 - **El fragment shader**

IDI_Q2 2018-2019



Processat de fragments: El fragment Shader

Fragment Shader

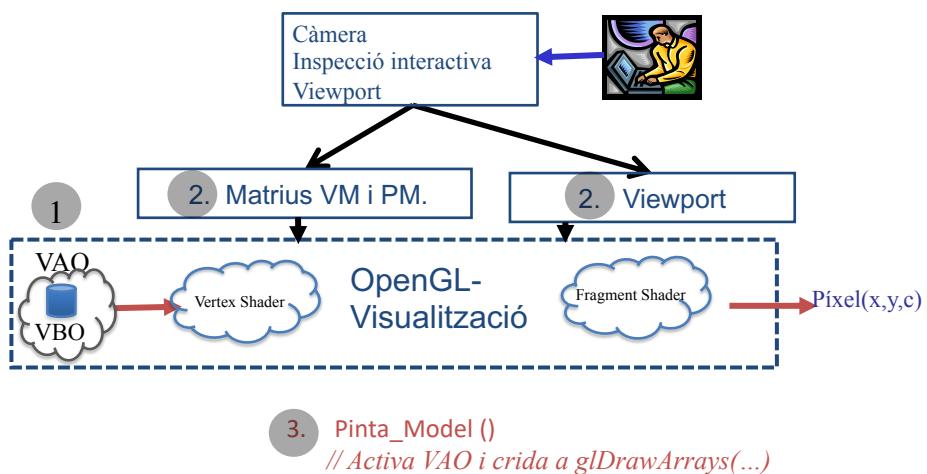
```
#version 330 core

out vec4 FragColor;

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

IDI Q2 2018-2019

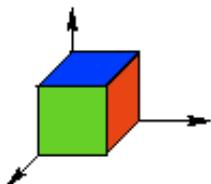
Pintar/visualitzar en OpenGL 3.3 (resum)



IDI Q2 2018-2019

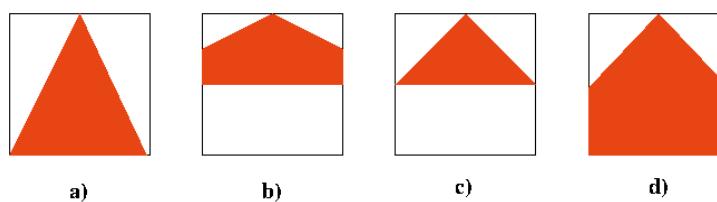
Exemple 2. Tenim una escena amb un cub de costat 2 orientat amb els eixos i de manera que el seu vèrtex mínim està situat a l'origen de coordenades. La cara del cub que queda sobre el pla $x=2$ és de color vermell, la cara que queda sobre el pla $z=2$ és de color verd i la resta de cares són blaves.

- a) Indica TOTS els paràmetres d'una càmera perspectiva que permeti veure complertes a la vista només les cares vermella i verda. La relació d'aspecte del viewport (vista) és 2. Fes un dibuix indicant la imatge final que s'obtindria.
 b) Quin efecte tindria en la imatge final modificar l'òptica ortogonal? Defineix la càmera ortogonal.



Exemple 3

Tenim una escena amb un triangle vermell amb vèrtexs $V1=(-2,0,0)$, $V2 = (2, 0, 0)$ i $V3=(0, 1, 0)$. Suposant que tenim un viewport quadrat de 600x600 pixels, i que hem inicialitzat les matrius de càmera (view) i projecció (proj) a la matriu identitat, indica quina de les següents imatges és la que sortirà en un viewport de 600x600 (sabem que el Vertex Shader i el Fragment Shader estan correctament implementats):



Classe 3: Contingut

- Especificació de càmera
- Exemple
- El procés de visualització projectiu (breu repàs)
- Processament de vèrtexs:
 - Seqüència de processos/etapes
 - Matrius requerides
 - El vertex shader
- Processament de fragments:
 - El fragment shader

IDI Q2 2018-2019