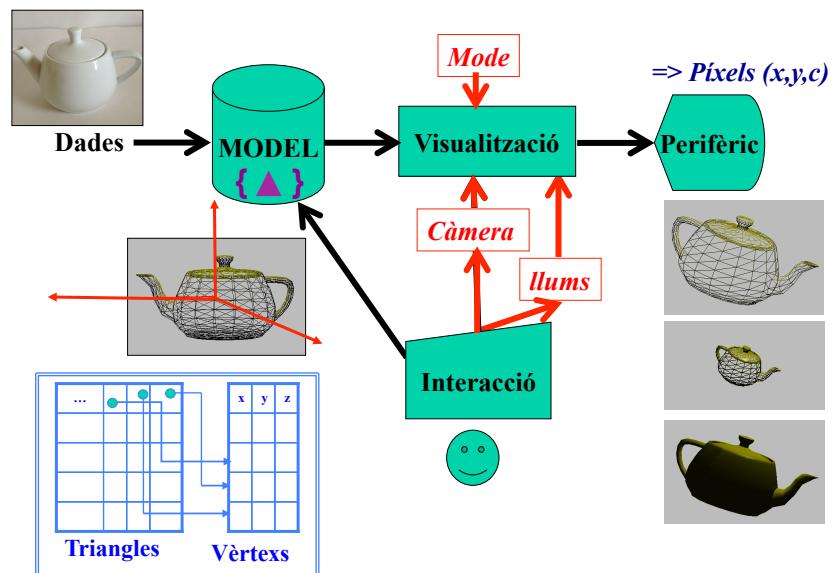


Classe 2: contigut

- Procés de Visualització (1)
- Models geomètrics (2): Escenes
- Breu repàs de TG i primers exercicis de TG

Visualització (intro)



Com indicar càmera?

The diagram shows a 3D scene with a camera on a tripod. A bounding box labeled "Volum visió" represents the field of view. Inside this volume, two points are defined: "OBS" (Observer) at the top right and "VRP" (Viewing Reference Point) at the bottom left. A "window" is indicated by a dashed line. Two coordinate systems are shown: the "Sistema Coordenades Aplicació/món/escena" (Scene Coordinate System) with axes x_A , y_A , and z_A ; and the "Sistema Coordenades Observador" (Observer Coordinate System) with axes x_o , y_o , and z_o . The "UP" vector indicates the vertical direction.

1. Ubicació respecte SCA: obs, vrp, up
2. Definir Volum de Visió: òptica

IDI Q1 2016-2017 3

The diagram illustrates the camera rendering process:

1. Posició, orientació
2. Òptica
3. Fer la Foto
4. Emmarcar

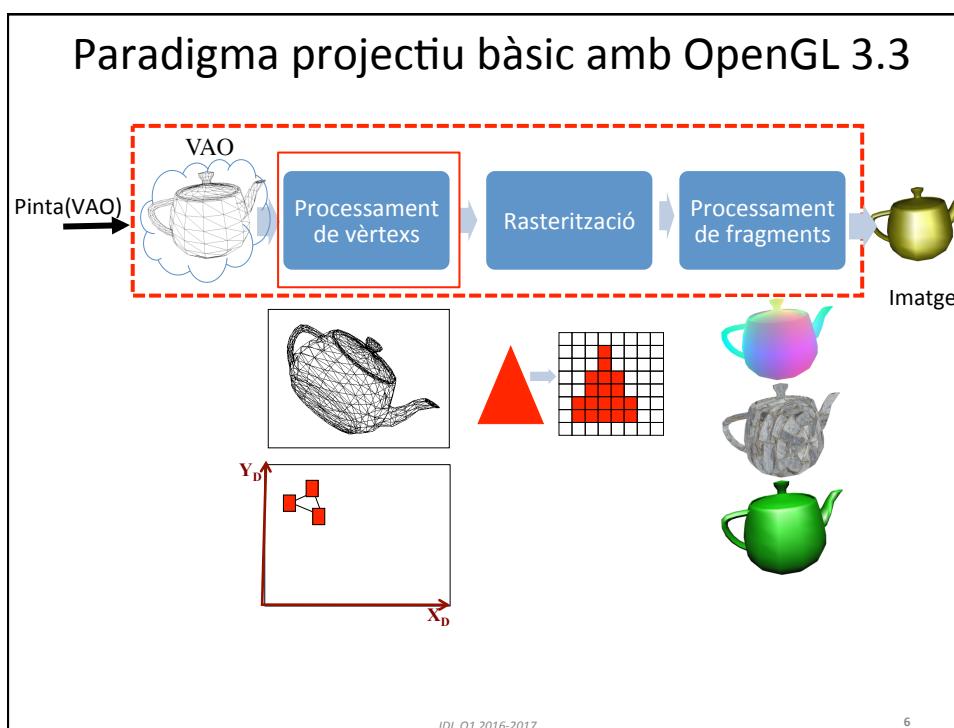
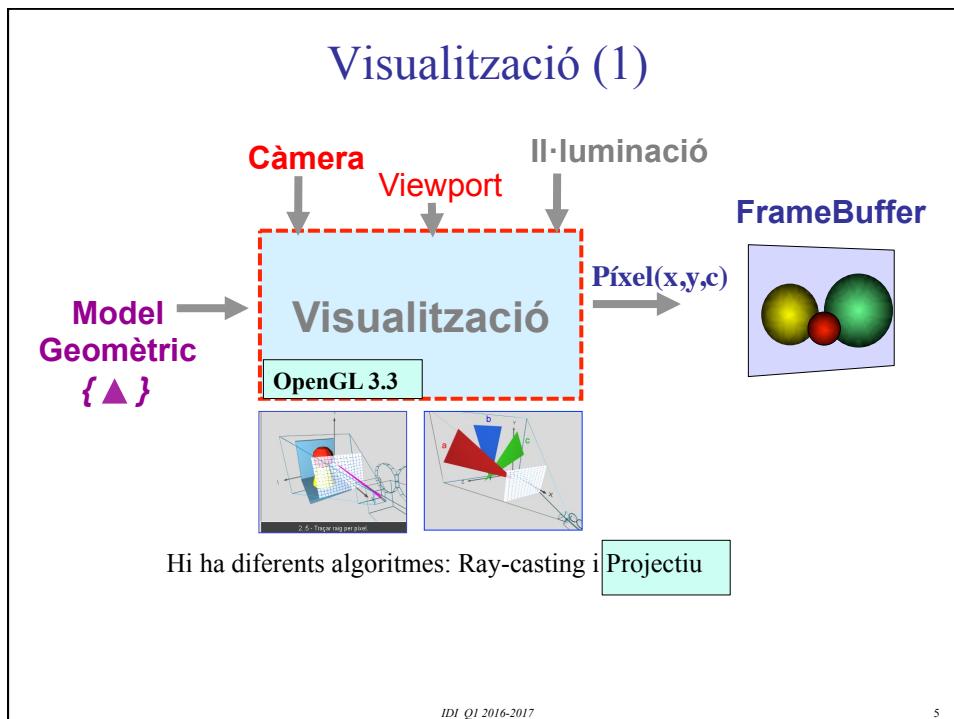
A green arrow points from the camera setup to a rendered image of Albert Einstein's head. Below, a diagram shows the OpenGL windowing process:

- Finestra OpenGL (Pantalla)**: A large gray rectangle representing the screen.
- Viewport/Vista**: A smaller red-bordered rectangle within the screen, representing the rendered image area.
- Coordinates**: Axes x_D and y_D are shown, indicating the position and size of the viewport.

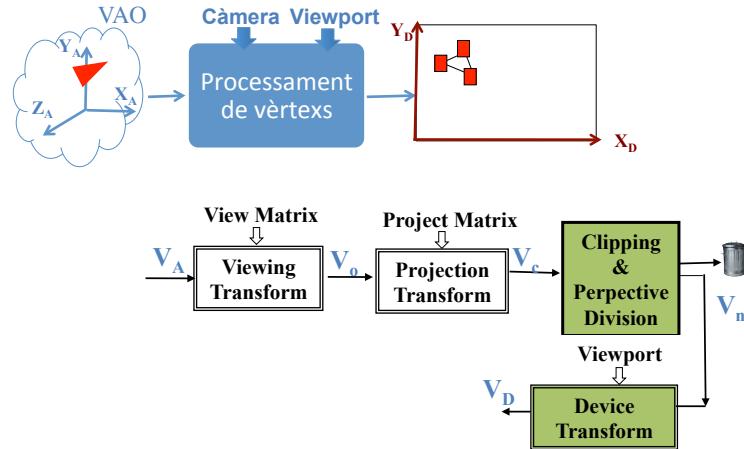
Usualment:

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

IDI Q1 2016-2017 4



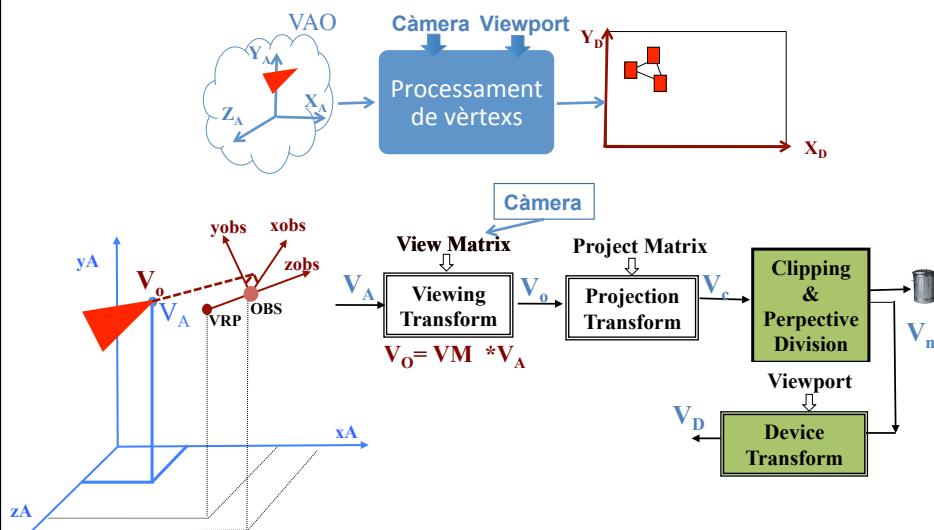
Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

7

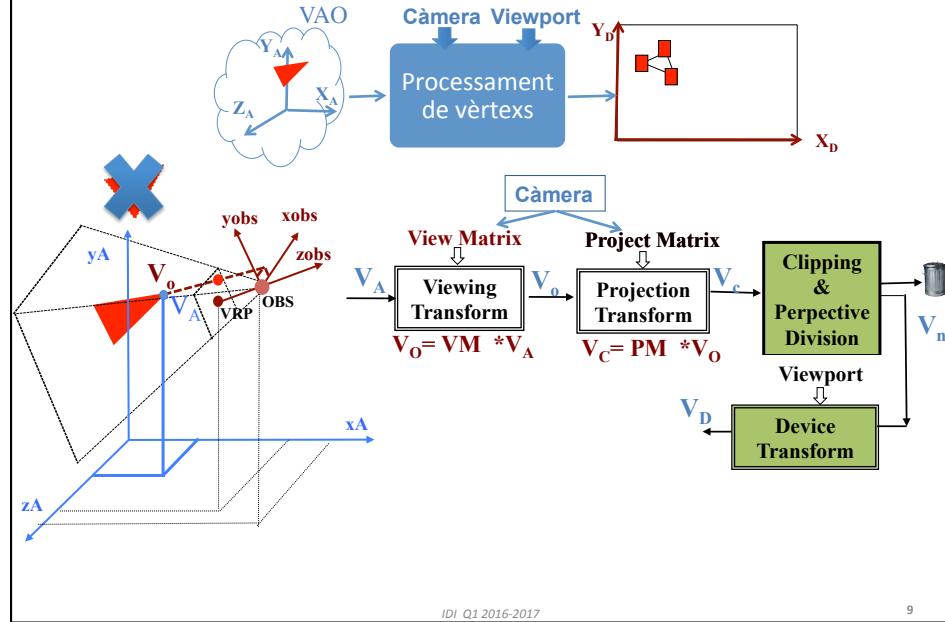
Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

8

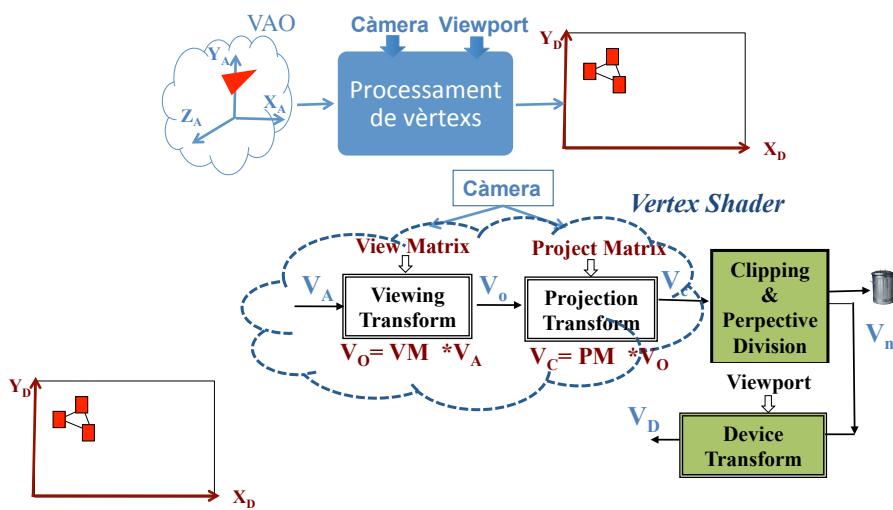
Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

9

Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q1 2016-2017

10

Paradigma projectiu bàsic amb OpenGL 3.3

Vertex Shader

```
#version 330 core

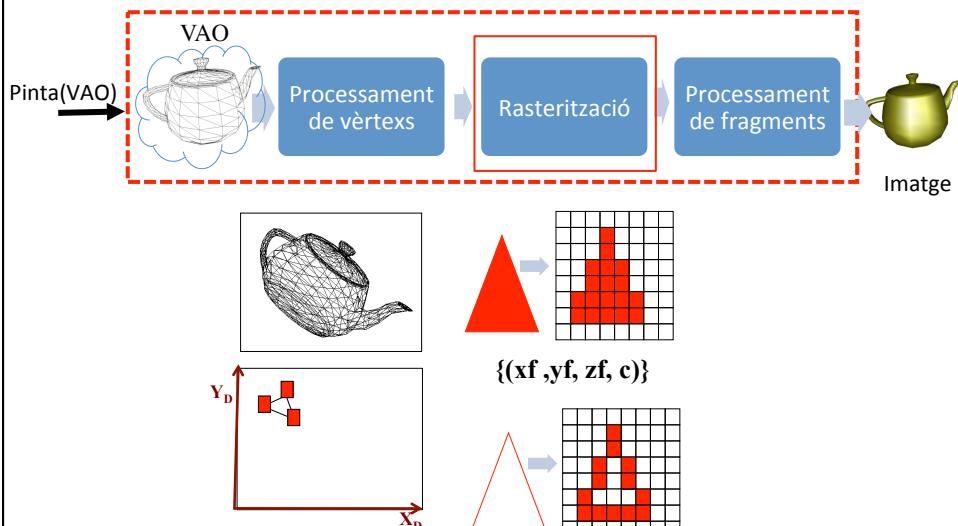
in vec3 vertex;
uniform mat4 PM;
uniform mat4 VM;

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

IDI Q3 2016-2017

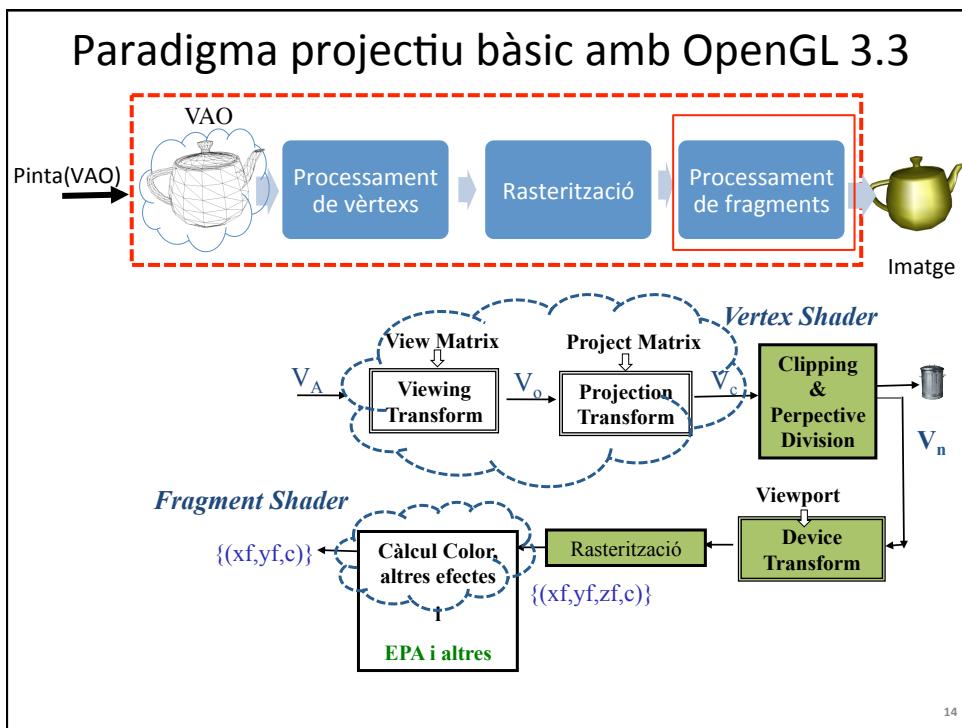
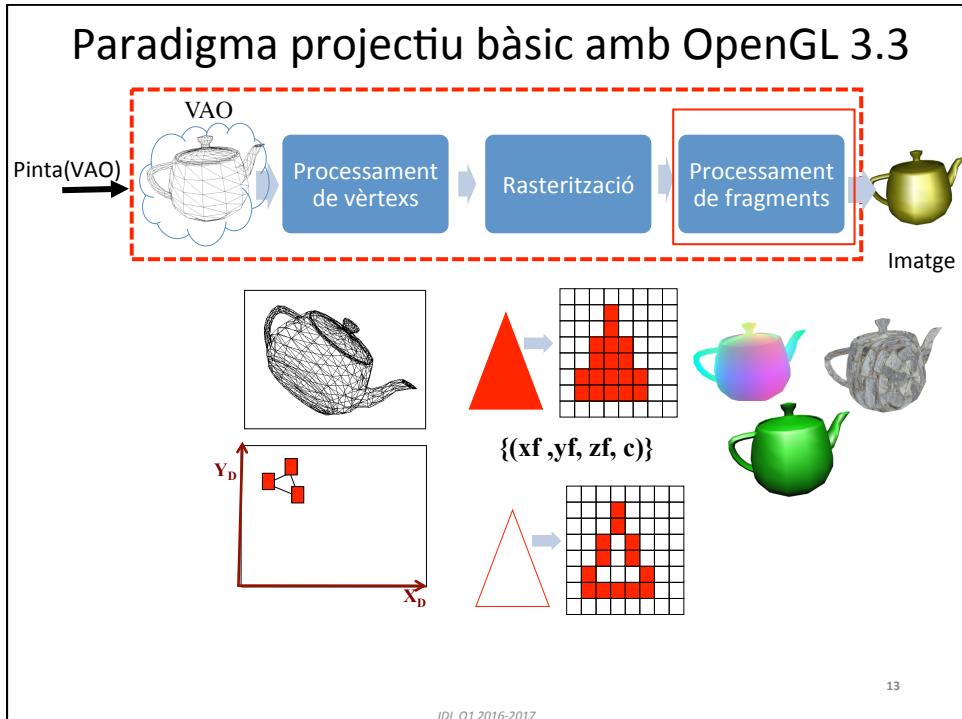
11

Paradigma projectiu bàsic amb OpenGL 3.3



IDI Q3 2016-2017

12



Fragment Shader

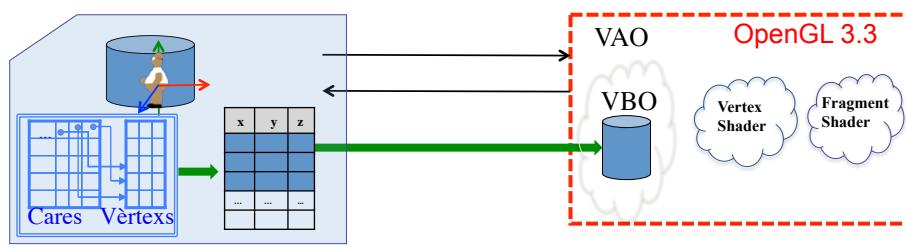
```
#version 330 core

out vec4 FragColor;

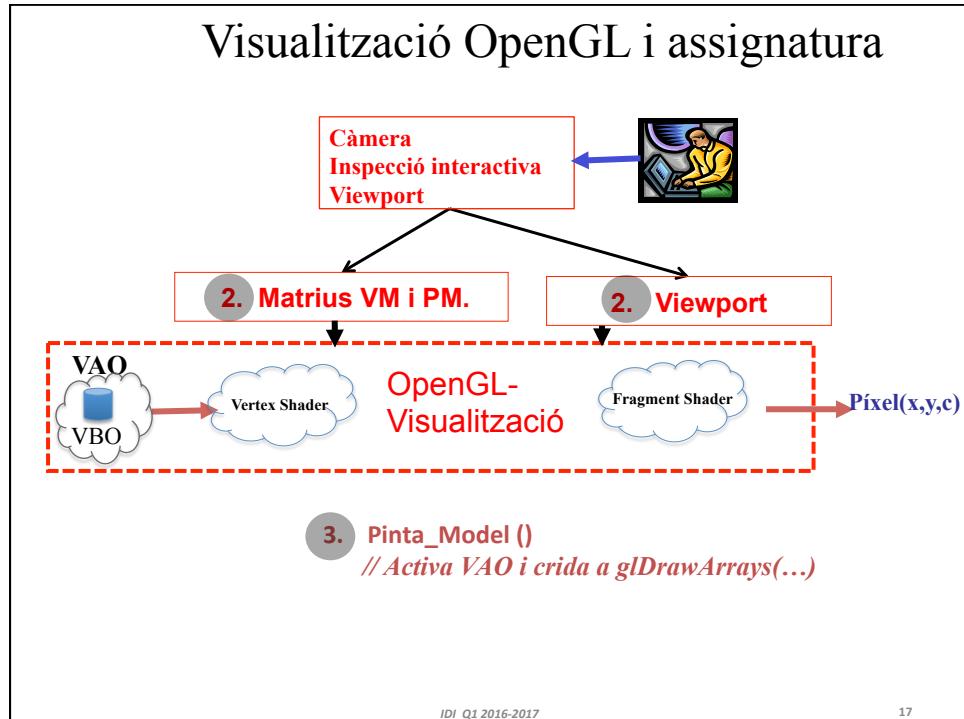
void main() {
    FragColor = vec4(0, 0, 0, 1);
}
```

Visualitzar en OpenGL 3.3: “core” mode

1. Generar VAO per cada model 3D que tinguem a memòria (NOMÉS cal fer-ho un cop si no modifiquem el model).



Aplicació. Model Geomètric



IDI Q3 2016-2017

17

Classe 2: contigut

- Visualització (1)
- Models geomètrics (2na part): Escenes
- Breu repàs de TG i primers exercicis de TG

IDI Q1 2016-2017

18