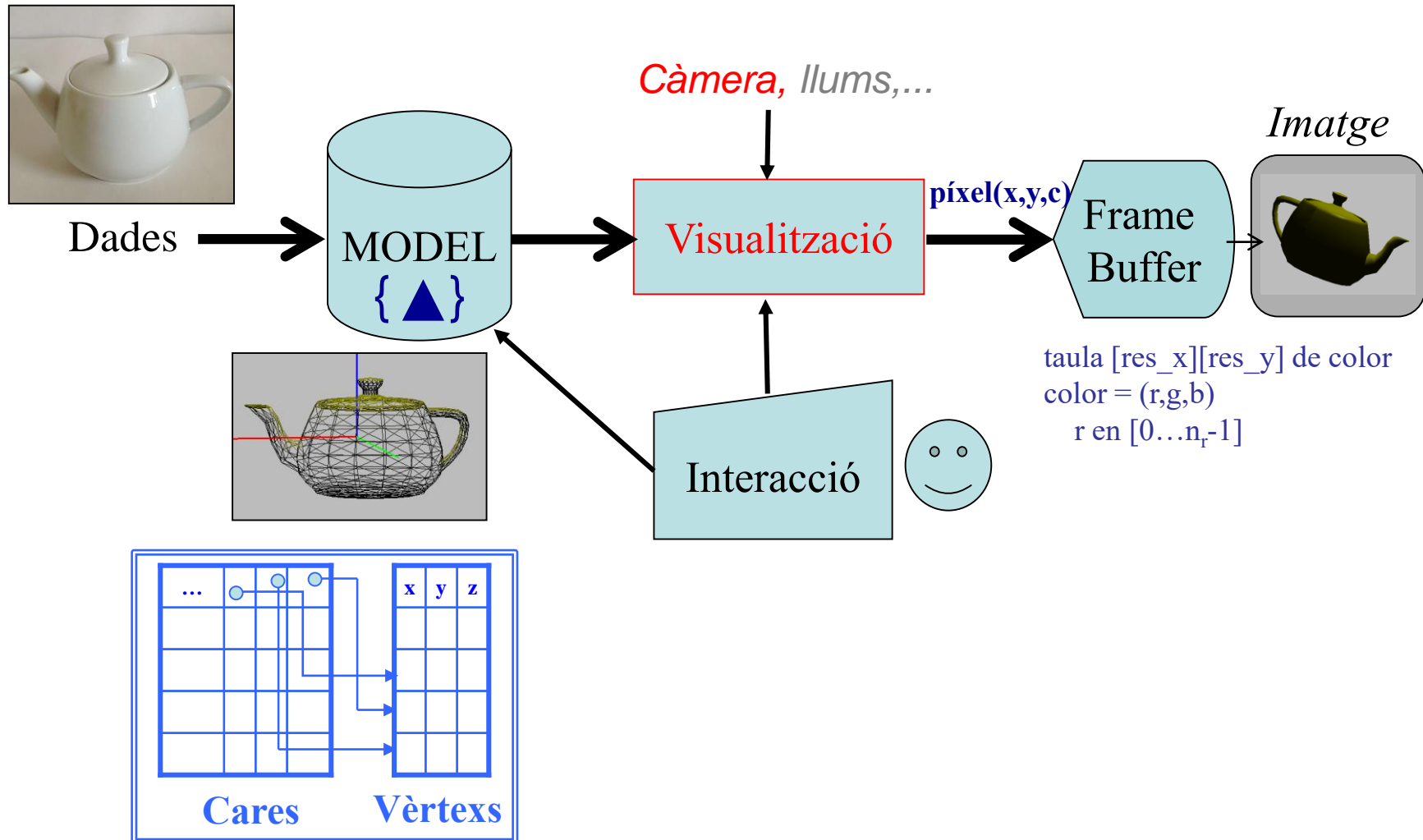
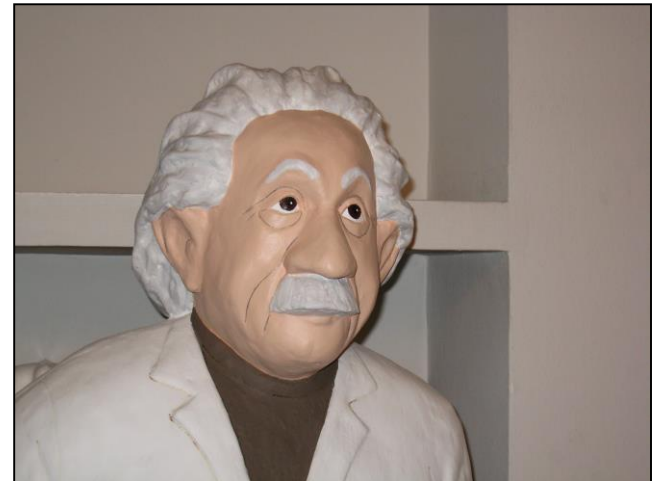
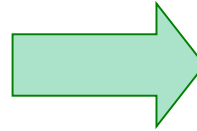


# Classe 2: Contingut

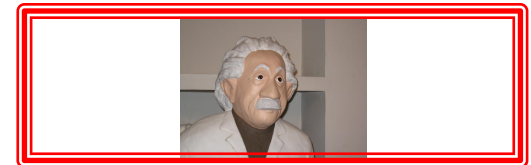
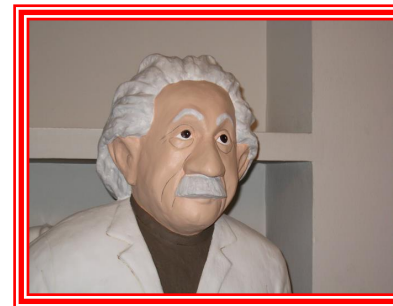
- **Introducció al procés de visualització**
- Transformacions geomètriques
- Exercicis

# Visualització: Introducció





**4. Emmarcar**



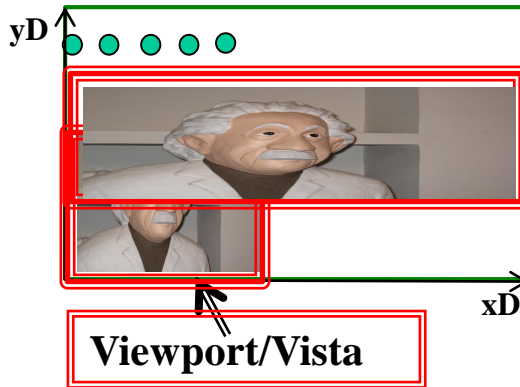
- Usualment, el viewport és tota la finestra OpenGL
- De moment, no ens preocuparem de si hi ha “deformacions”

**1. Posició, orientació**

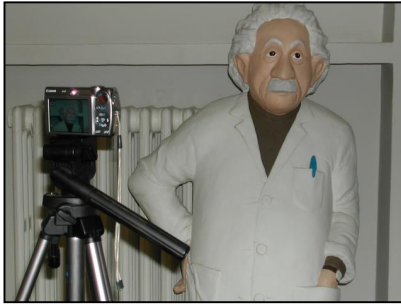
**2. Òptica**

**3. Fer la Foto**

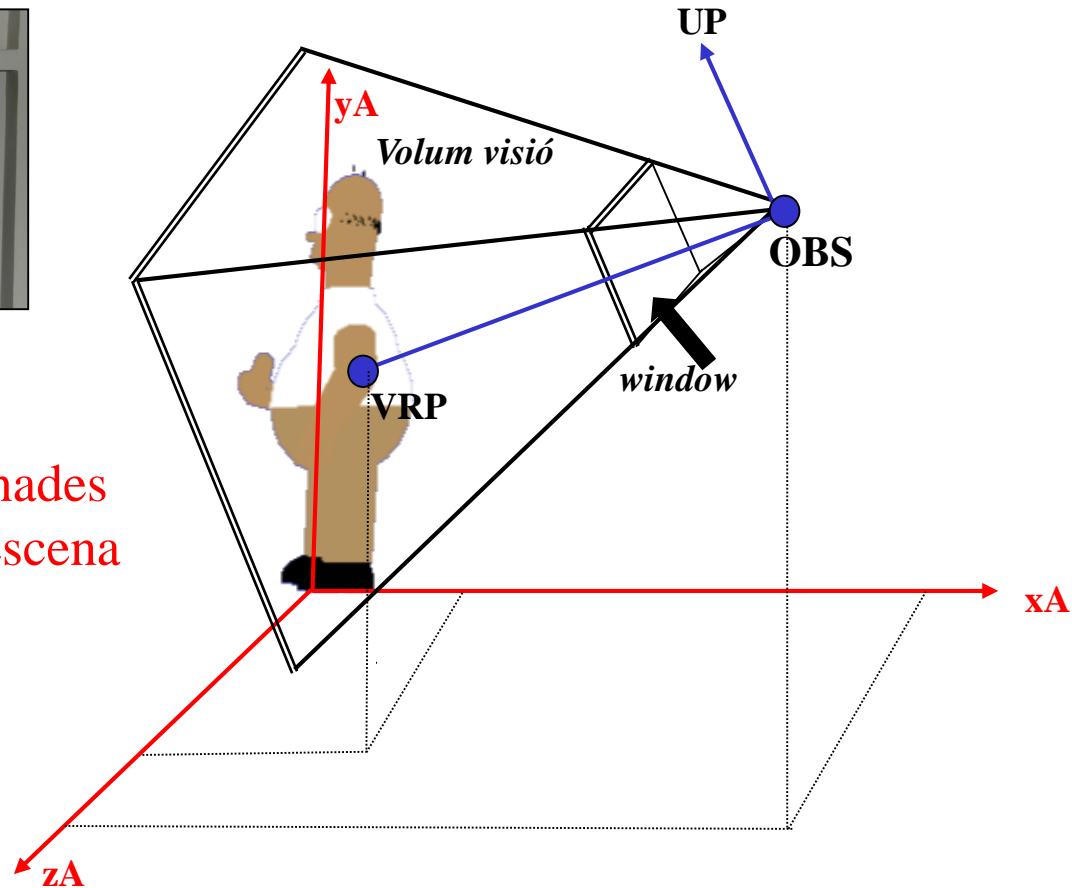
**Finestra OpenGL (Pantalla)**



# Com indicar la càmera?

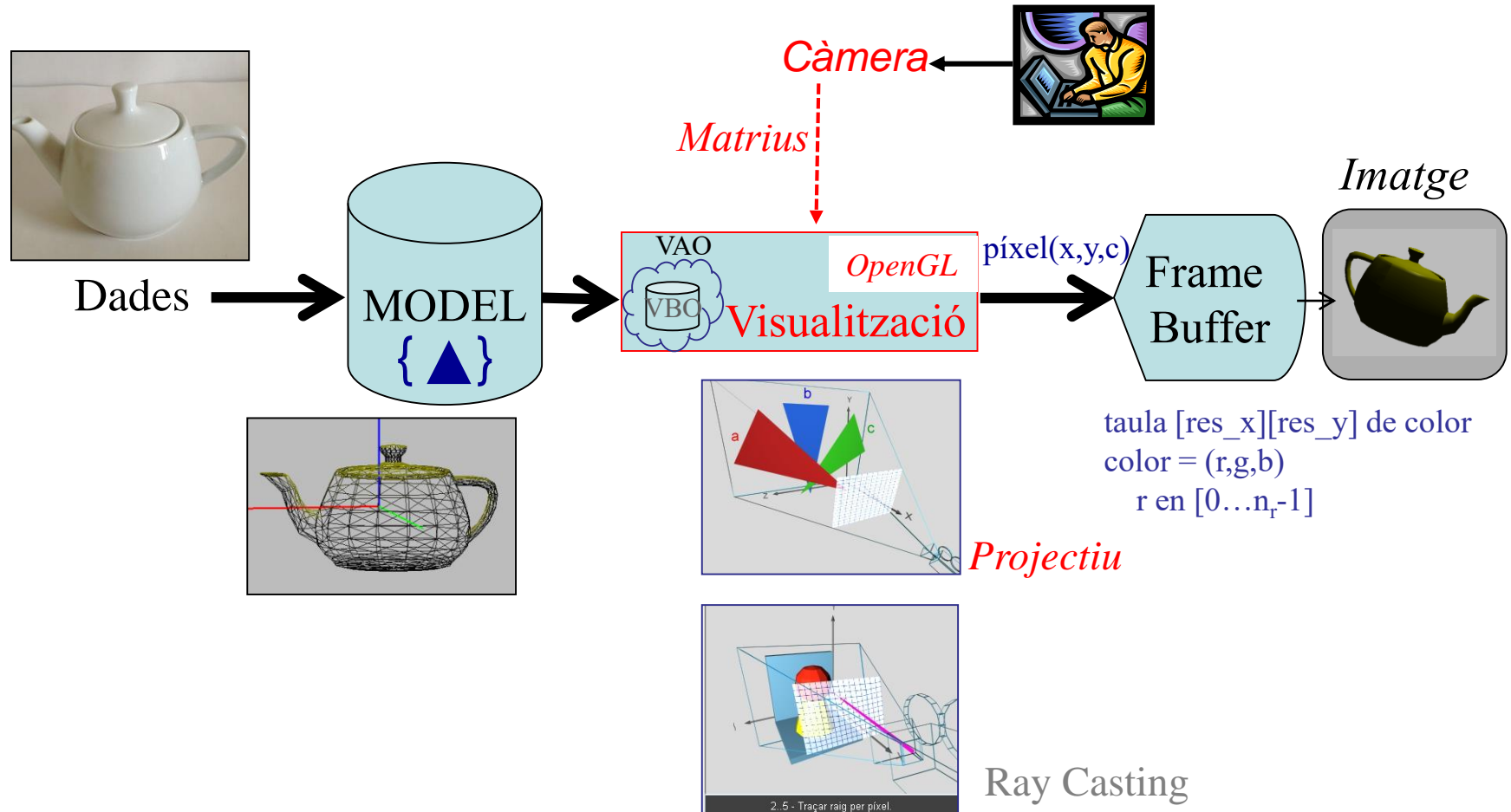


Sistema Coordenades  
Aplicació/món/escena

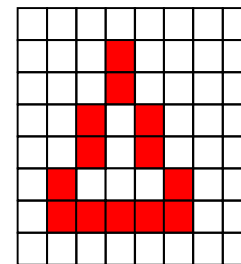
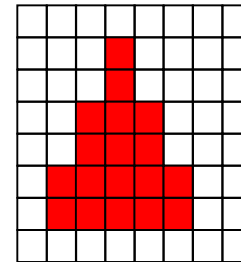
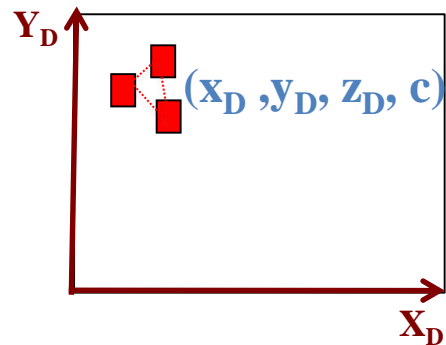
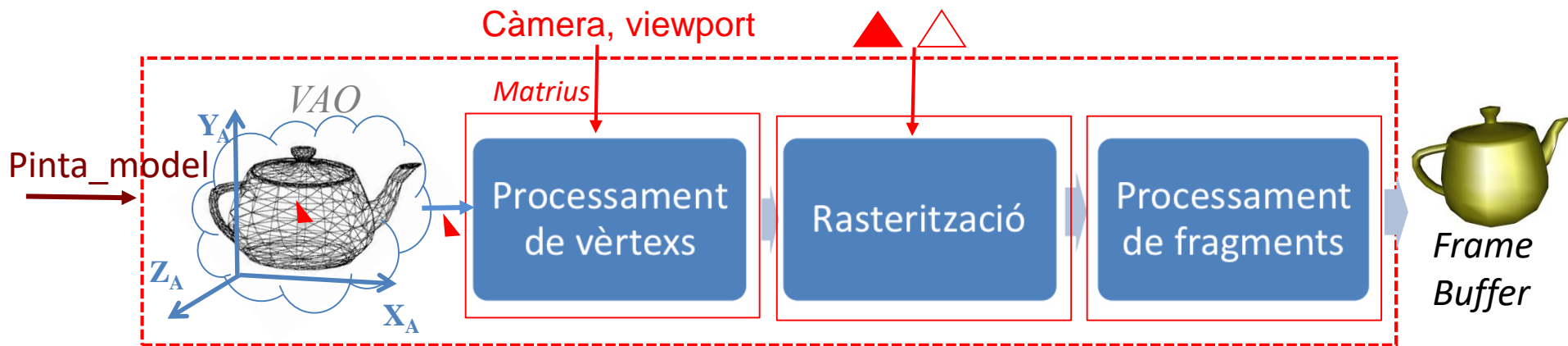
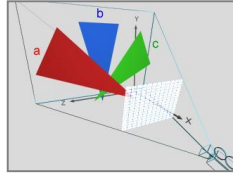


1. Ubicació respecte SCA: obs, vrp, up
2. Definir el Volum de Visió: òptica (window, zNear, zFar)

# Visualització: Introducció (2)



# Pintar/visualitzar en OpenGL 3.3 (1)

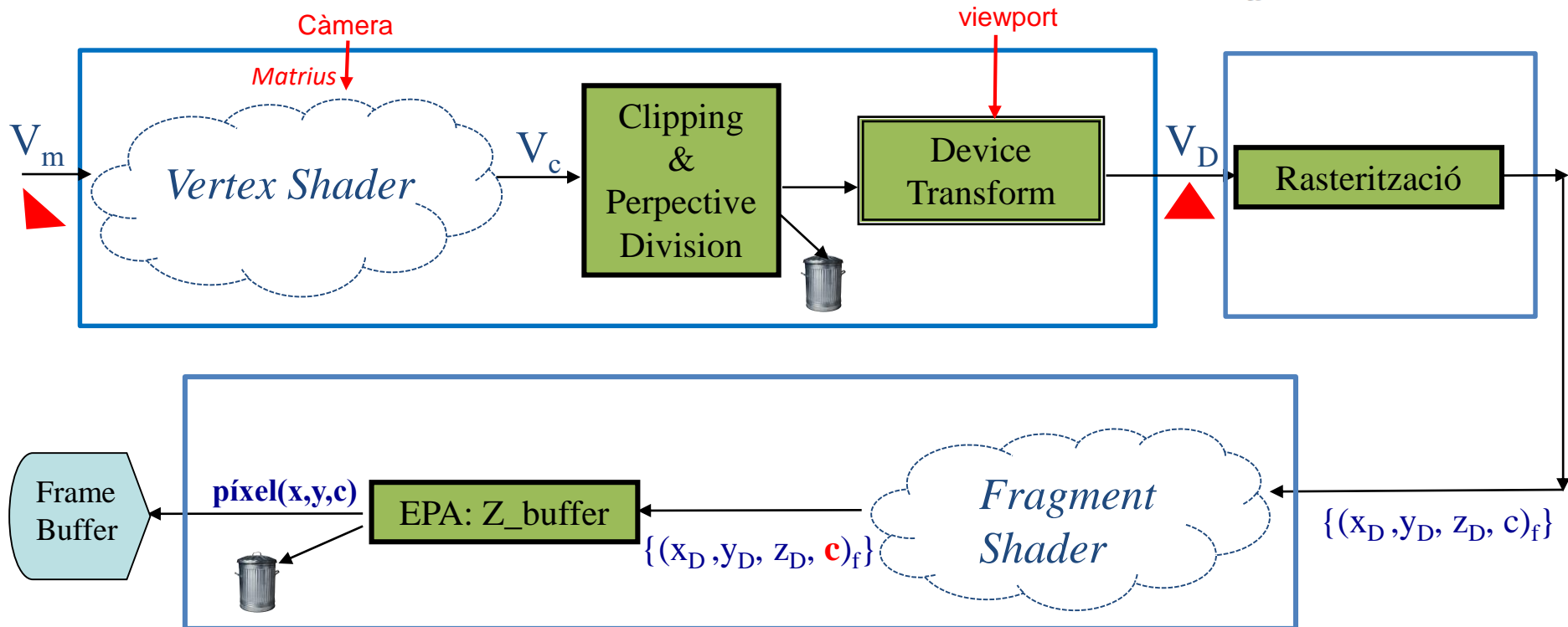
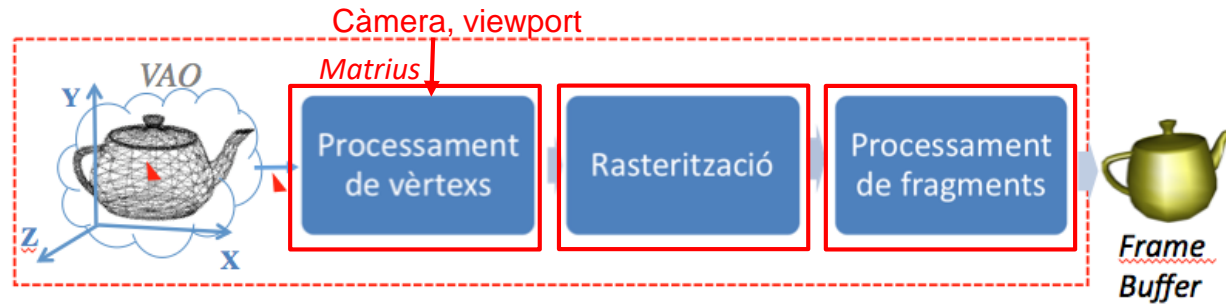
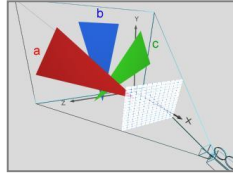


Fragments:  $\{(x_D, y_D, z_D, c)_f\}$



píxel(x,y,c)

# Pintar/visualitzar en OpenGL 3.3 (2)

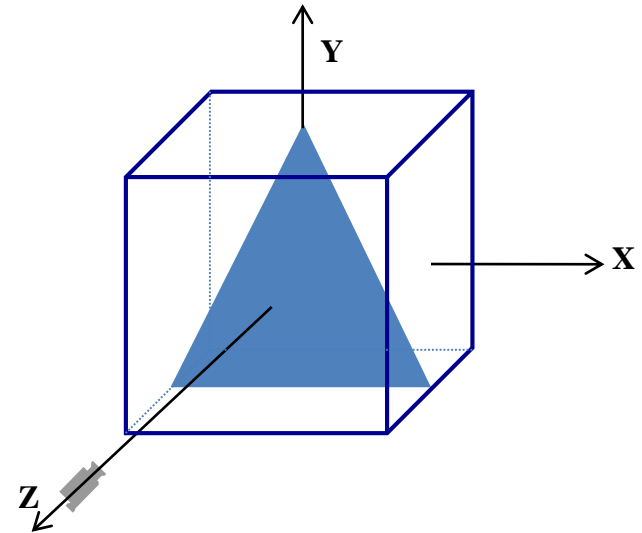


# Pintar/visualitzar en OpenGL 3.3 (3)

## *Vertex Shader*

```
#version 330 core
in vec3 vertex;

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

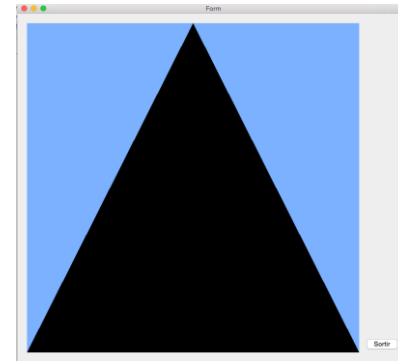


Volum de Visió cub de (-1,-1,-1) a (1,1,1)

## *Fragment Shader*

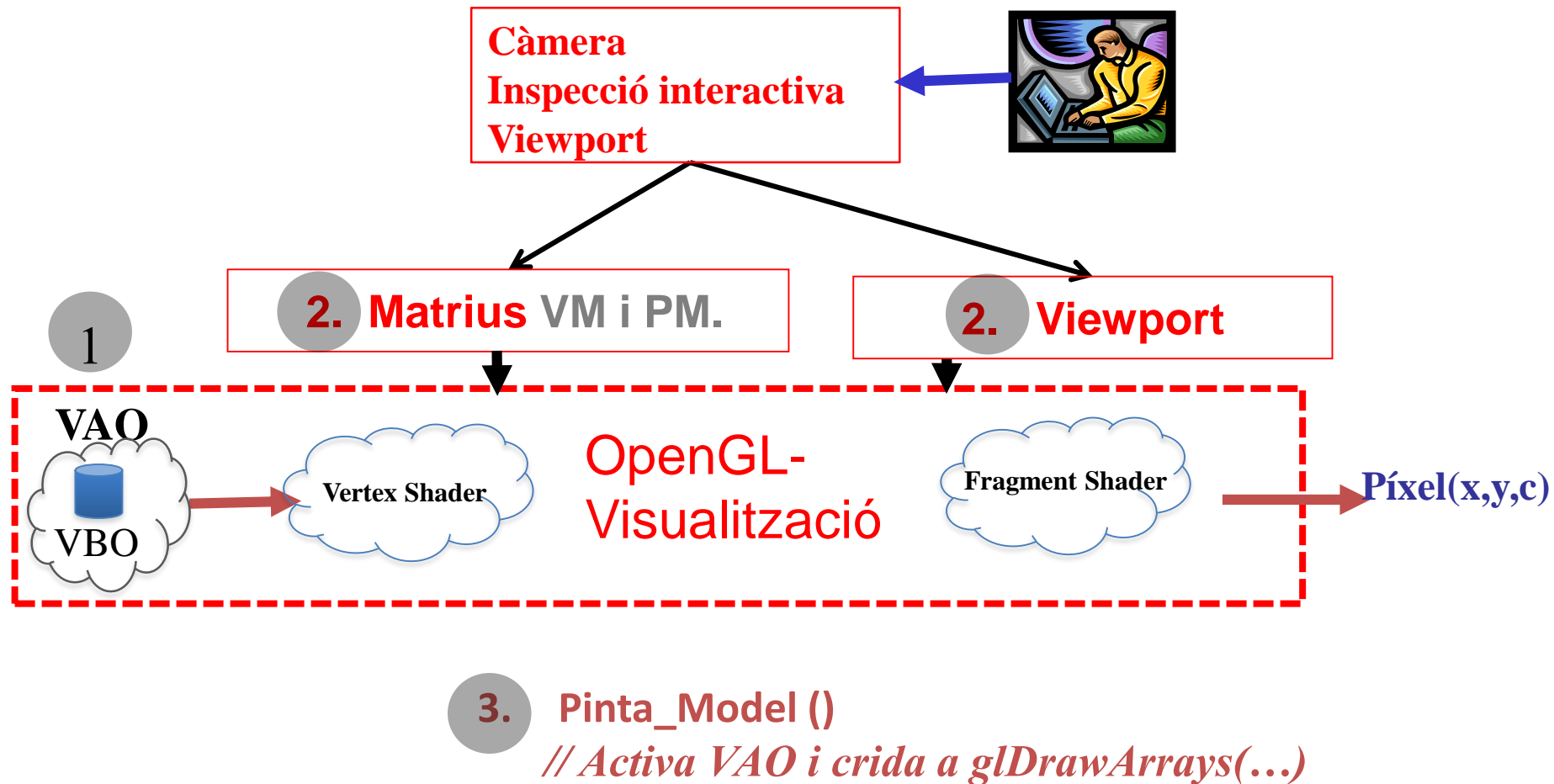
```
#version 330 core
out vec4 FragColor;

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





# Pintar/visualitzar en OpenGL 3.3 (resum)



# Classe 2: Contingut

- Introducció al procés de visualització
- Transformacions geomètriques
- Exercicis