centre adscrit a:
UNIVERSITAT DE BARCELONA

Computer Graphics

1. Introduction to computer graphics

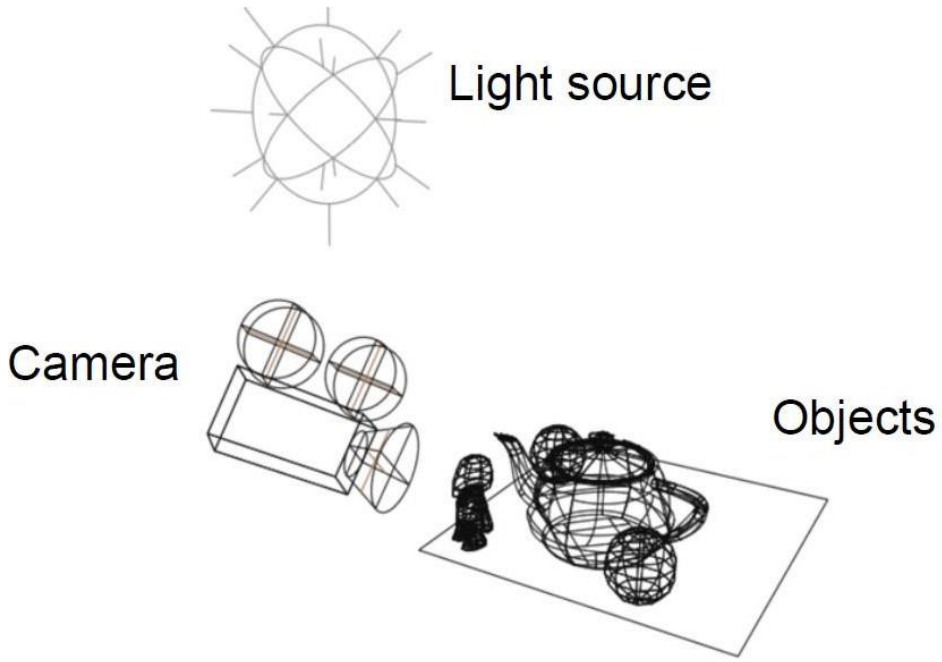
Dr Joan Llobera – joanllobera@enti.cat
Dr Jesús Ojeda – jesusojeda@enti.cat

Spring 2019

Outline

1. What is computer graphics?
2. Computer graphics model
3. Image Formation
4. The rendering pipeline

1. What is computer graphics?



1. What is computer graphics?

Definition:

- The process of creating images with a computer
- It involves hardware, software and applications

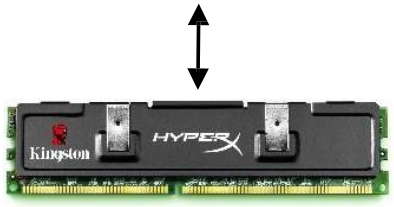
Applications

- Visualizing information
- Scientific Visualization
- Games
- Design
- User interfaces

2. Computer graphics model

Input devices

CPU



CPU memory

GPU

GPU memory

Framebuffer

Output device

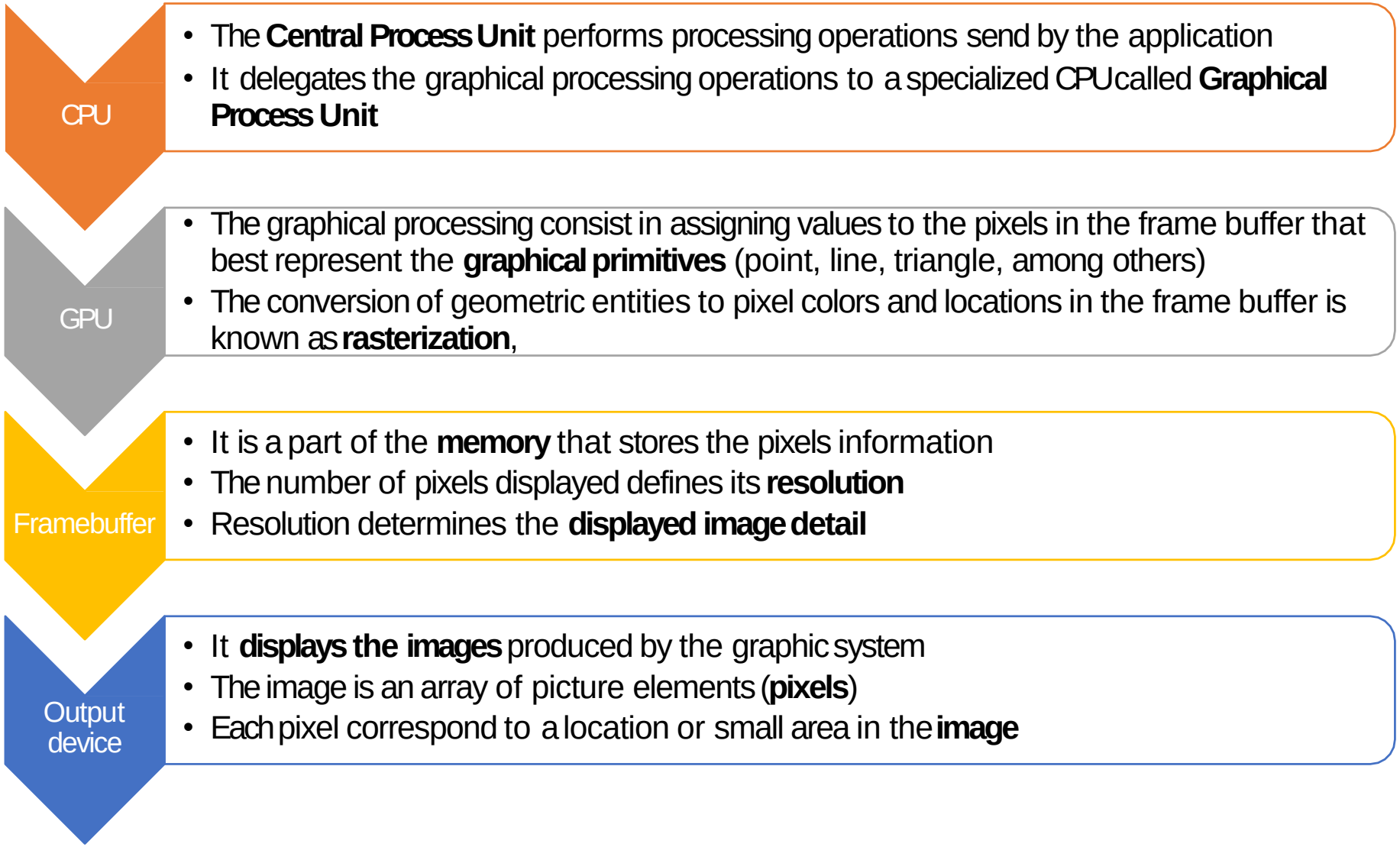
2. Computer graphics model

Examples:

- Workstations
- Laptops
- XBOX, Wii, playstation
- Mobile Phones
- …

CRT TV are not computer graphics systems

2. Computer graphics model



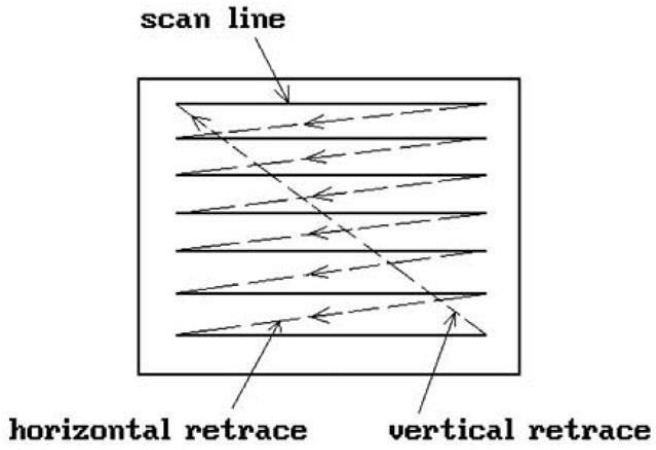
2. Computer graphics model

- The **frame buffer** is the core element and its main **characteristics** are:
  - The **resolution** is the number of pixels that will be drawn on the screen and determines the **detail of the image**
  - The **depth** (or also called the **precision**) defines the number of bits that are used for representing the information of a pixel and determines the **color displayed**
    Examples: 8 bit, HDR (up to 12 bits)

Source: https://www.cnet.com/news/hdr10-vs-dolby-vision-vs-hlg-how-do-hdr-formats-compare/

2. Computer graphics model

The **frame rate** (or also known as frame frequency and frames per second (FPS)) is the frequency (rate) at which an imaging device produces unique consecutive images called frames



The **refresh rate** is the number of times in a second that a display hardware updates its buffer.

- In a non-interlaced (or progressive) system, the pixels are displayed row by row, or scan line by scan line, at the refresh rate.
- In an interlaced display, odd rows and even rows are refreshed alternately

2. Computer graphics model

2. Computer graphics model

- The frame rate and the refresh rate are independent!
- Specific strategies need to be designed to synchronize both

2. Computer graphics model

Summary:

1. What does a computer graphics model look like
2. Input, CPU, GPU, Frame Buffer. Display
3. The Frame Buffer:
   - Characteristics (resolution+depth)
   - Frame rate and refresh rate

3.Image formation

We need to:

- Represent (and move) geometry
- Represent (and change) colour

The main elements are:

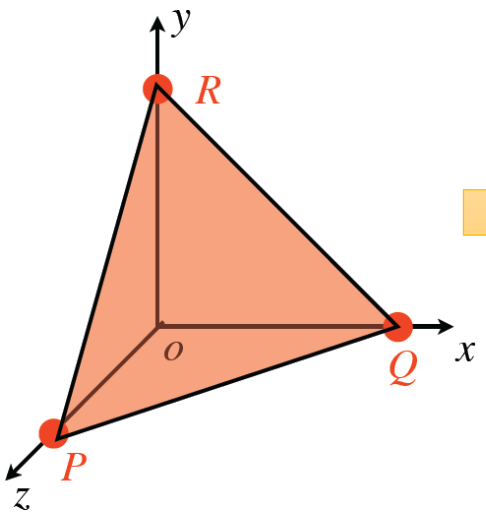
1. Objects
2. Viewer
3. Light Source

These three elements are independent!

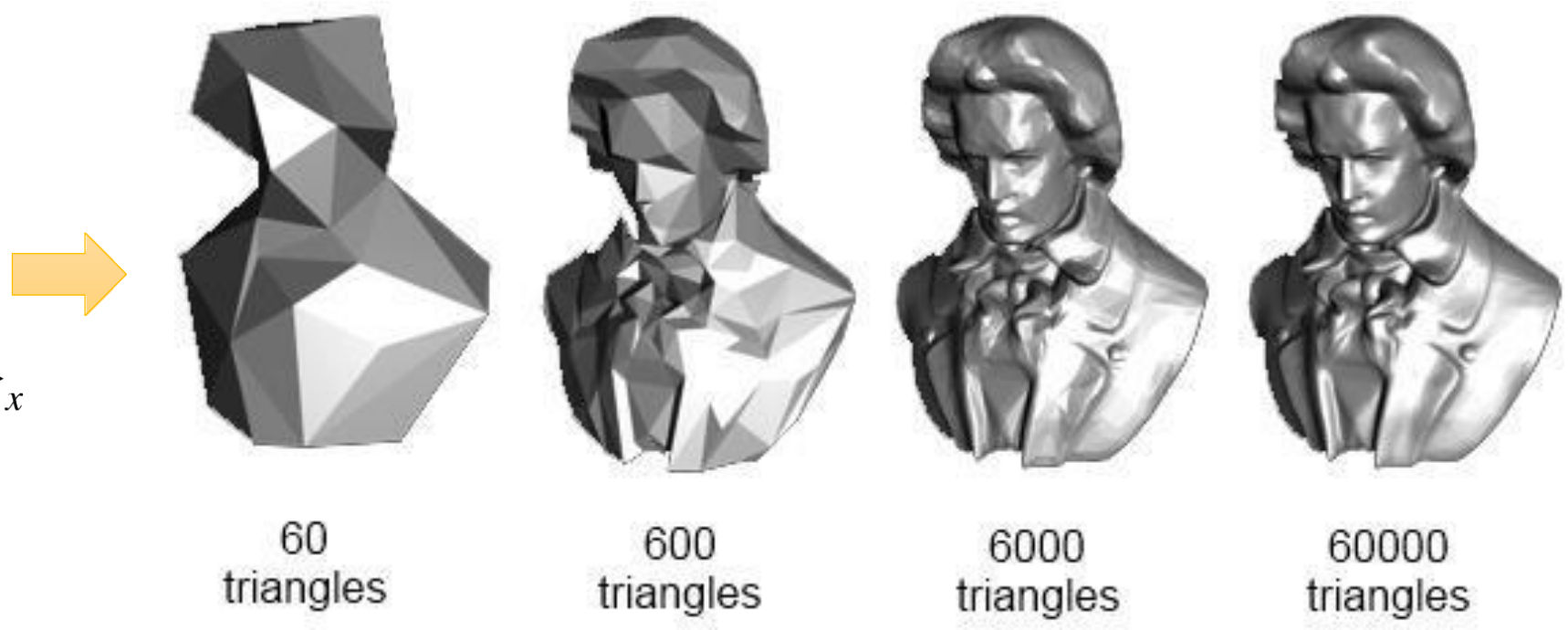
3 Image Formation

1. Objects
   - Formed of points, lines and polygons
   - Their existence is independent from the image-formation process
   - Specified with **vertices** and drawn with **triangles**

3 Image Formation



R, P and Q define a triangle composed by 3 vertices



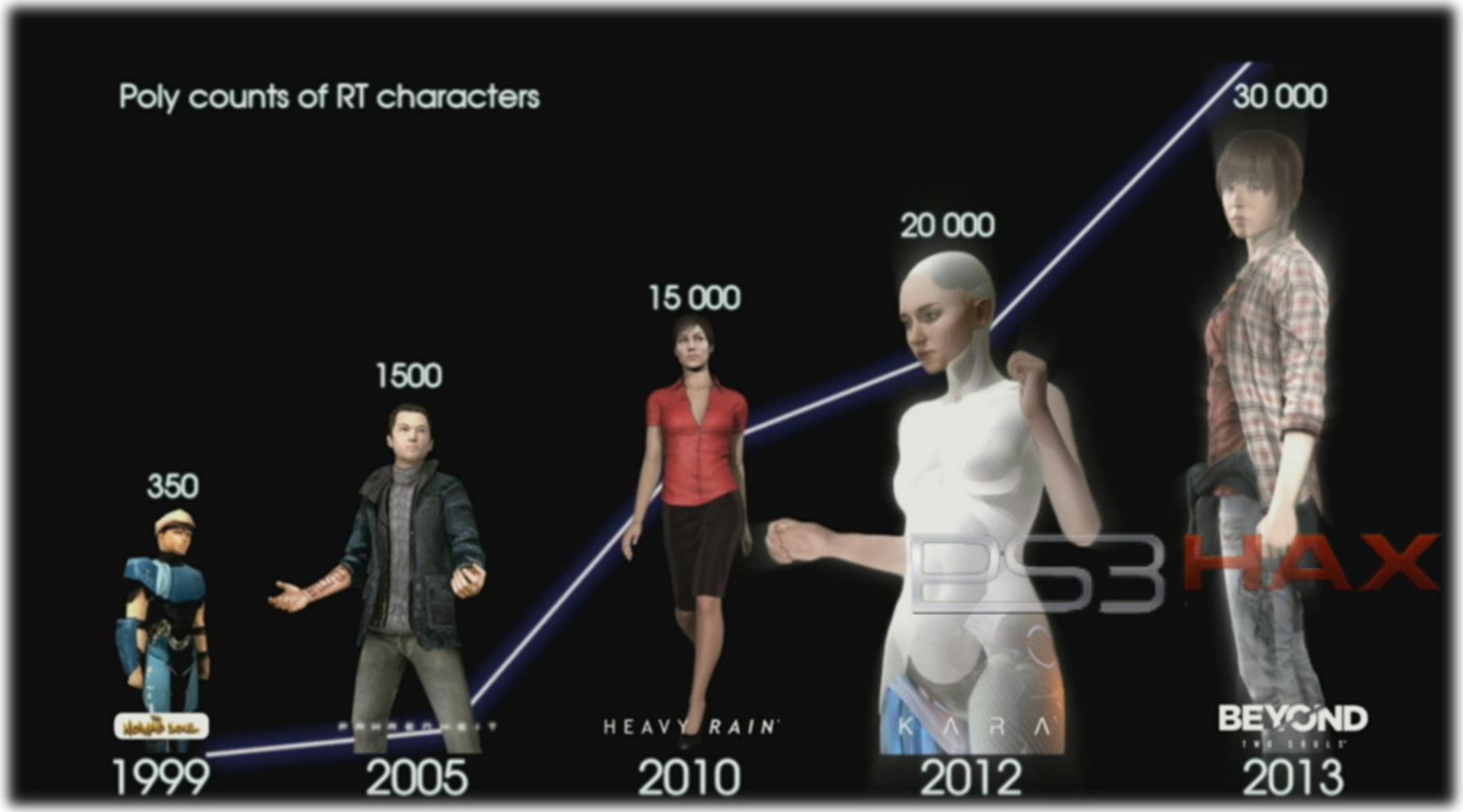
60 triangles

600 triangles

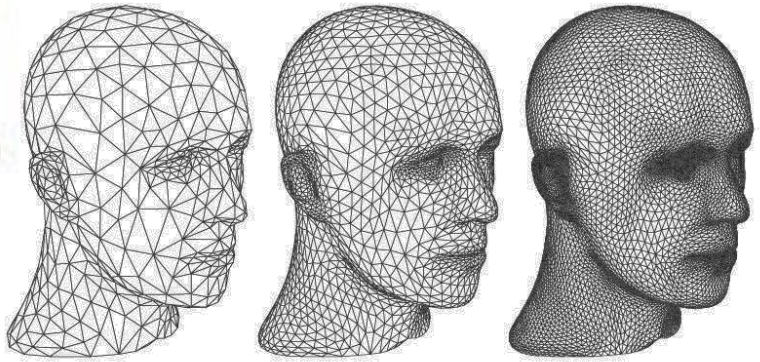
6000 triangles

60000 triangles

3 Image Formation



3 Image Formation

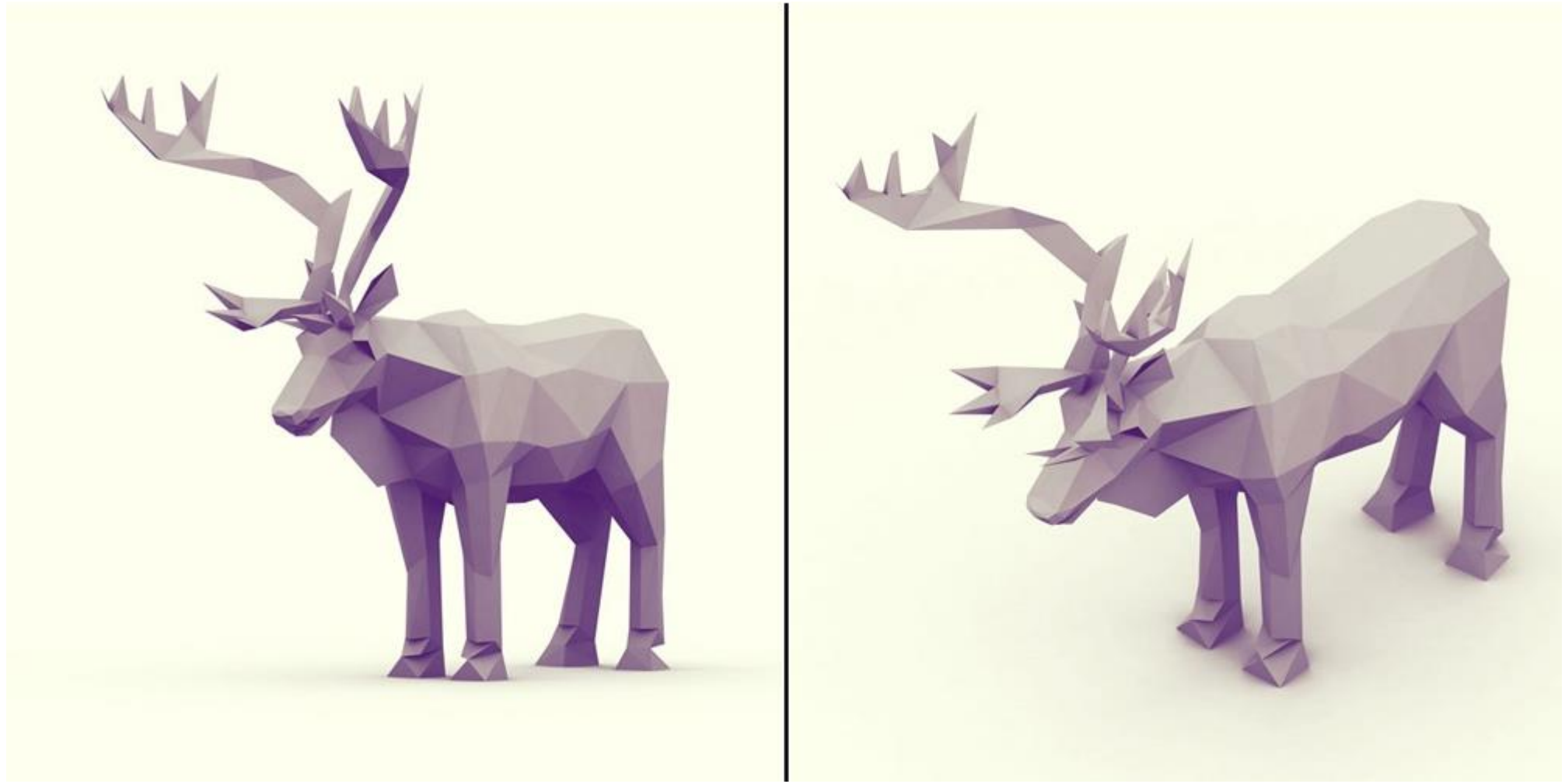


Original/Source
Common Model

Wii U
Max LOD Result

PS4/X720
Max LOD Result

3 Image Formation



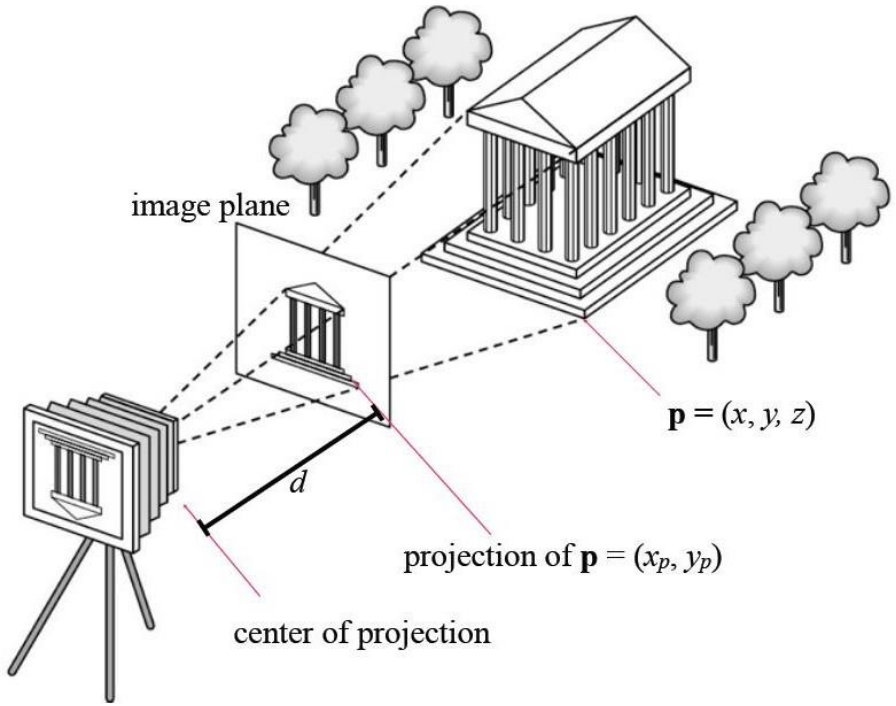
3 Image Formation

1. Objects
   - Formed of points, lines and polygons
   - Their existence is independent from the image-formation process
   - Specified with **vertices** and drawn with **triangles**

   - Character realism is associated with the number of polygons
   - The number of polygons matters
   - However, low-poly can also be creatively interesting

3 Image Formation

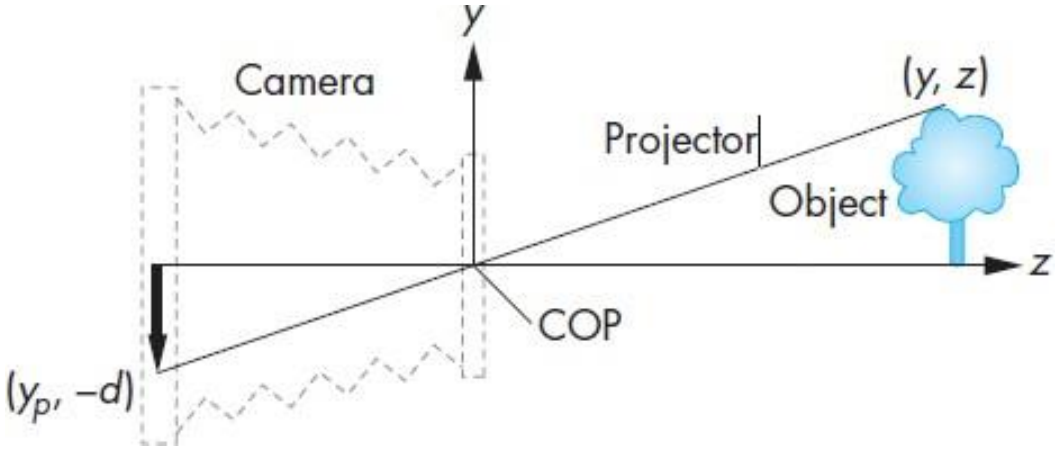
2. Viewer

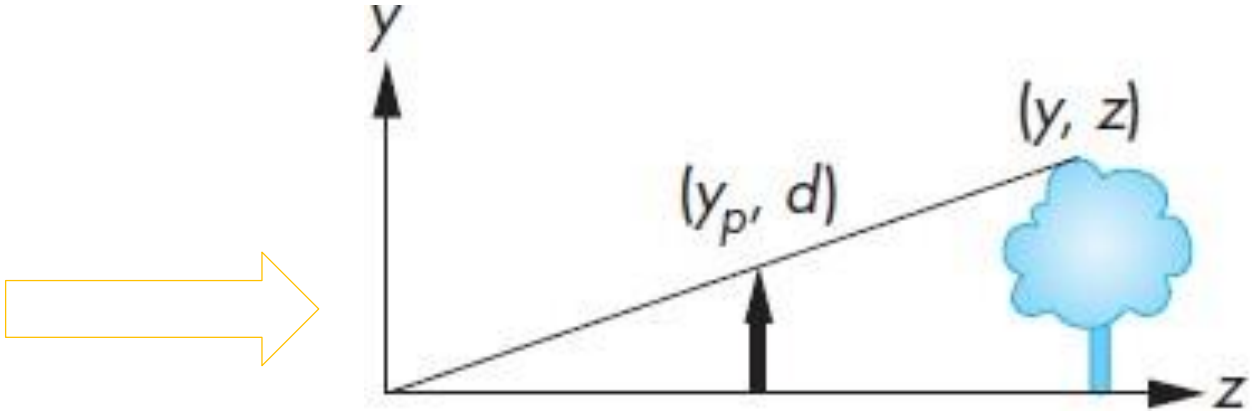


1. The objects interact but the viewer only films
2. The camera has a position, an orientation and a vision field

3 Image Formation

2. Viewer

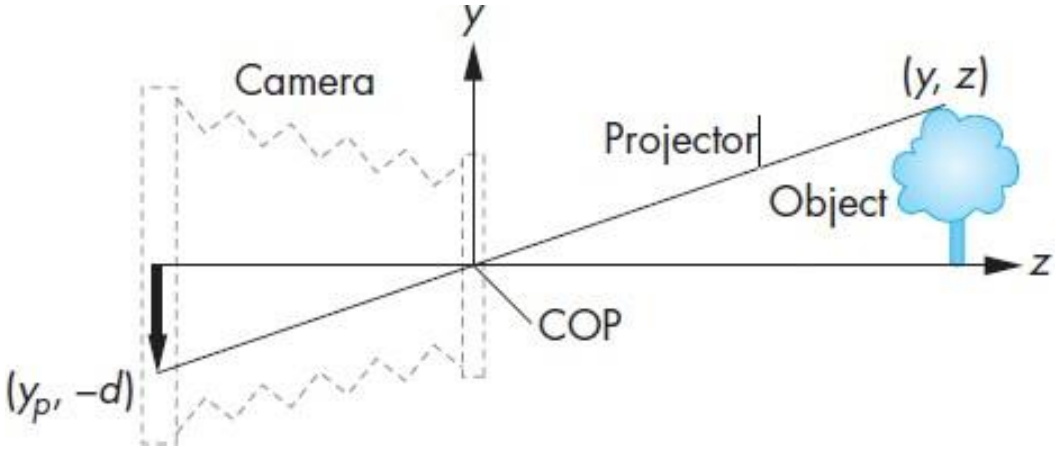


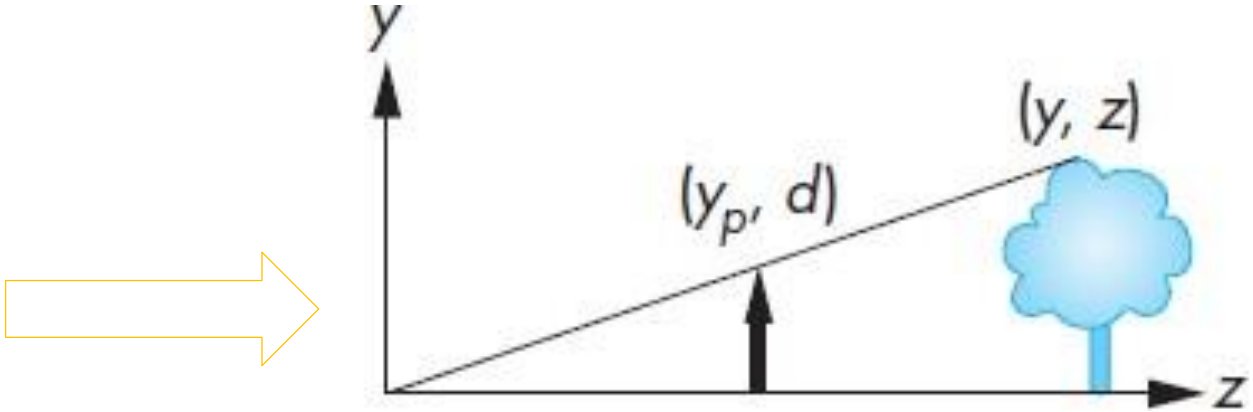


$$\tan \propto = \frac{y}{z} = \frac{y_p}{d}$$

3 Image Formation

2. Viewer



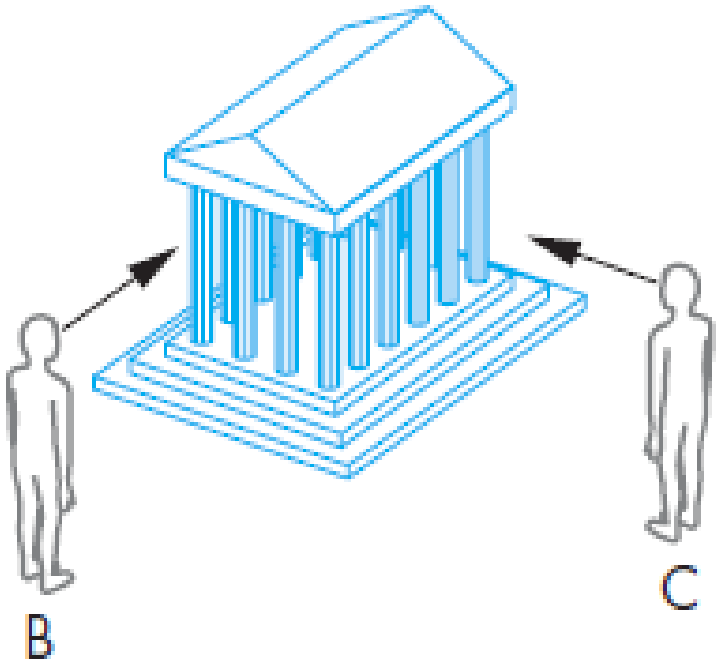


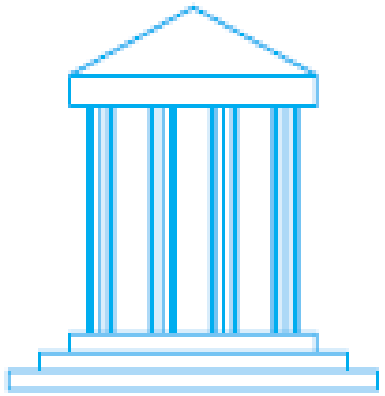
Perspective

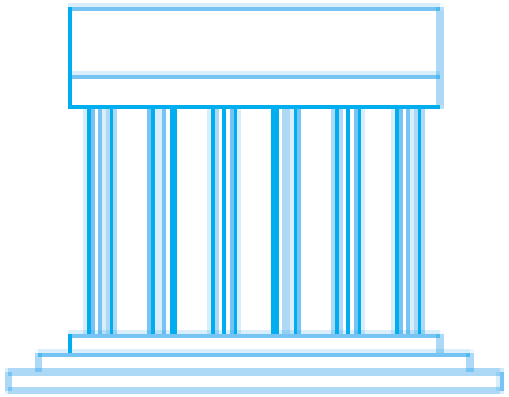
$$\tan \propto = \frac{y}{z} = \frac{y_p}{d}$$

3 Image Formation

2. Viewer







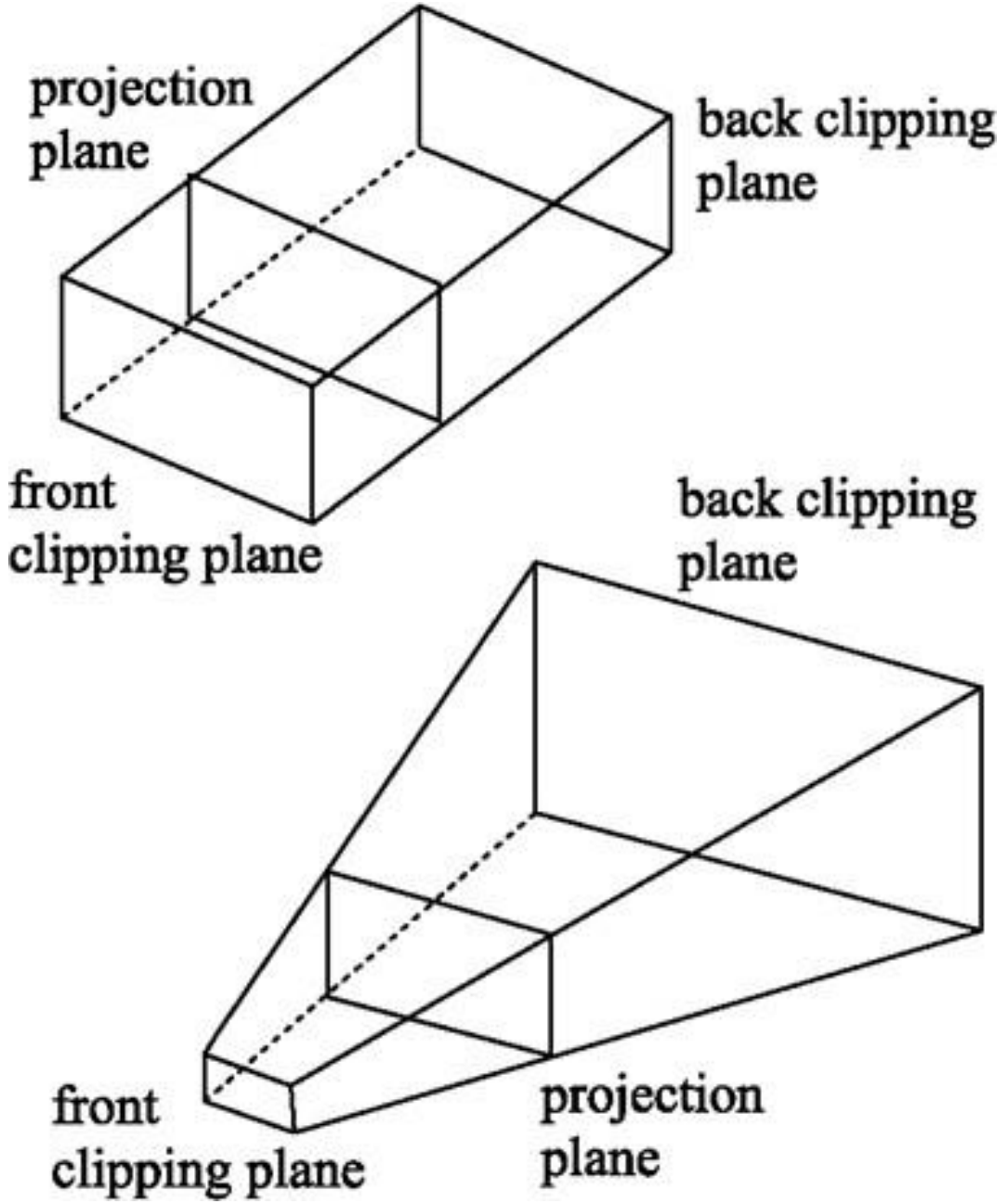
Orthographic

3 Image Formation

Orthographic

vs

Perspective



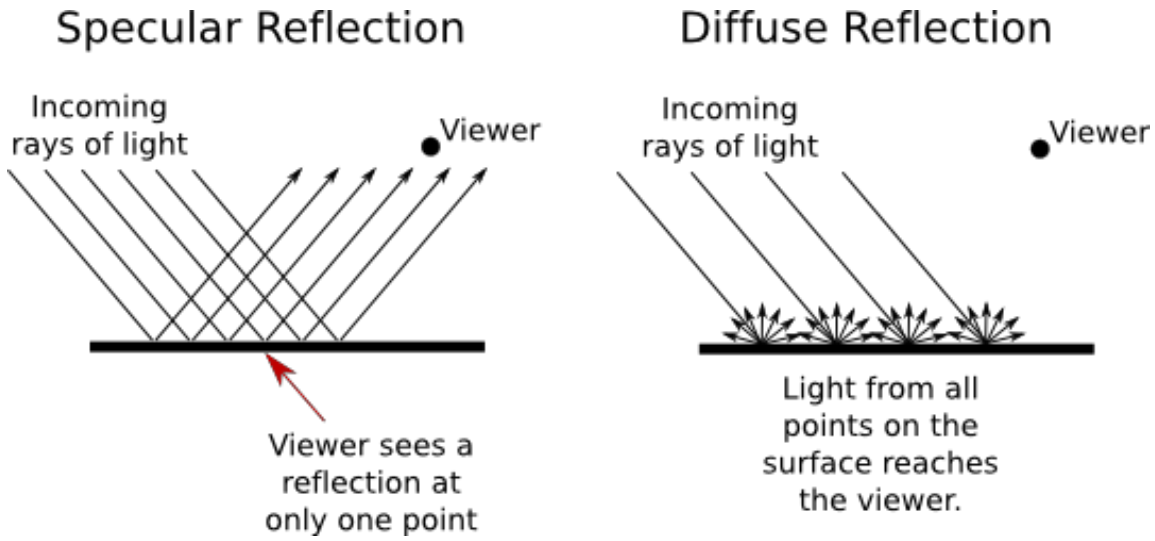
3 Image Formation

3. Lighting

An object is covered by a material which has, at least:

- **Absorption**: color properties
- **Scattering:** Diffuse and/or specular

Intensity and direction of light, plus the materials determines result

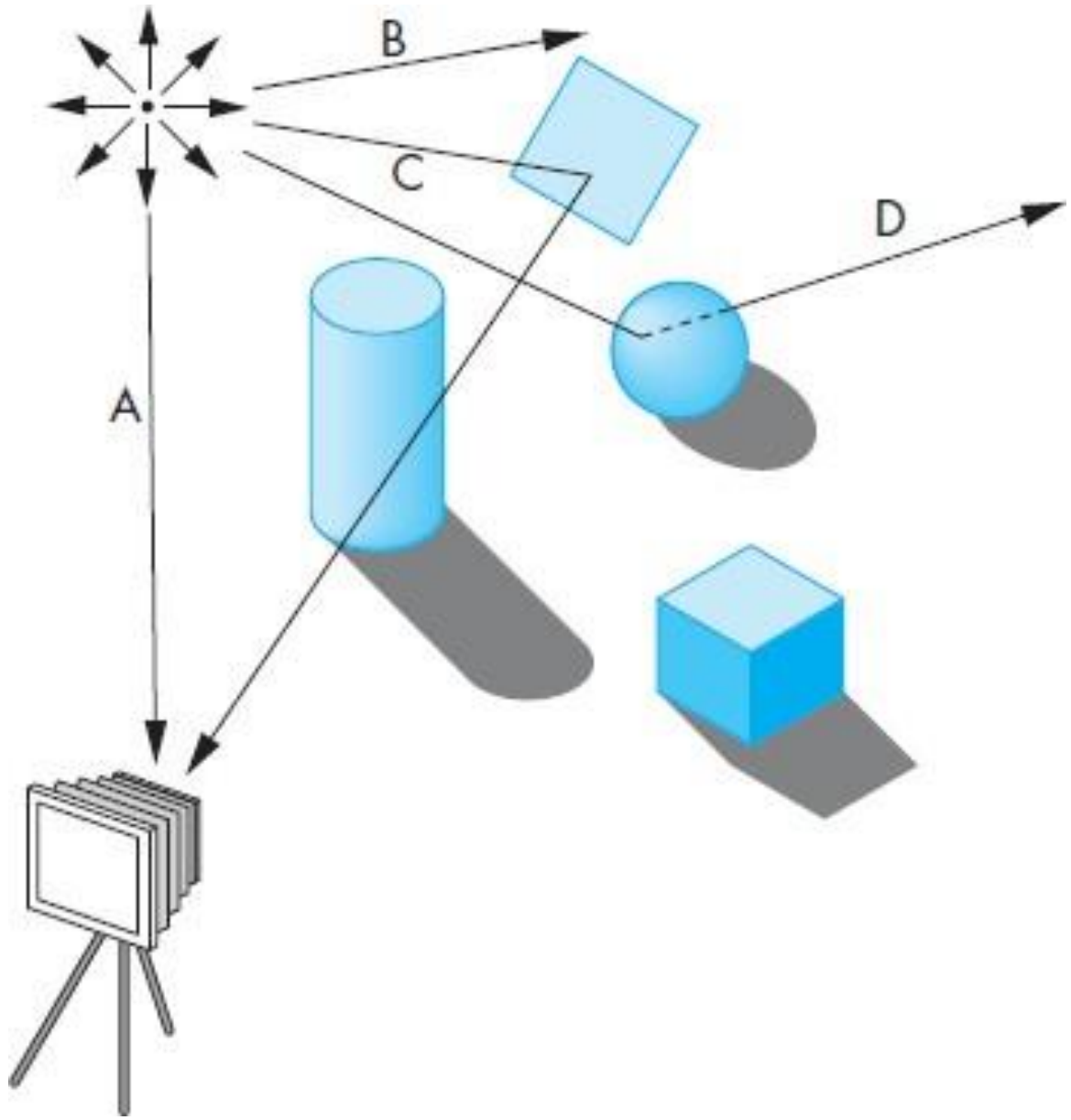


3 Image Formation

3. Lighting

We can consider:

- Only direct illumination
- Direct and indirect illumination

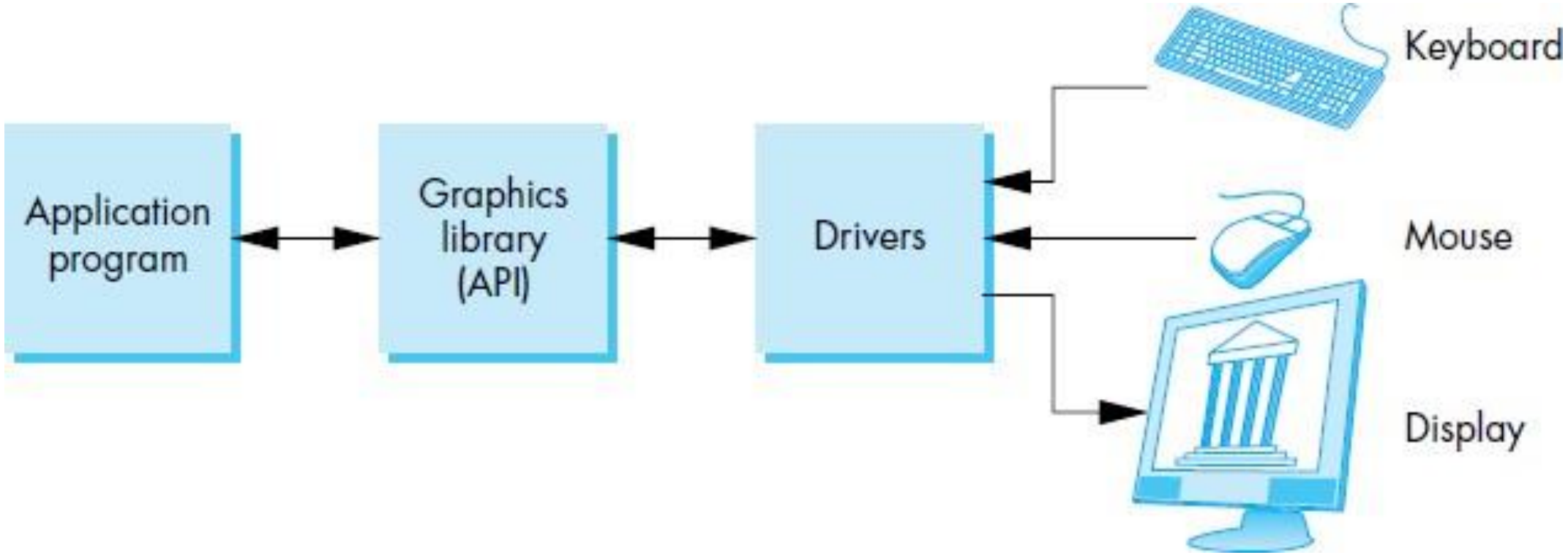


3.Image formation

Summary:

1. **Objects**
   are pixels and triangles
2. **Viewer** is a perspective or orthographic projection
3. **Lighting** is defined by
   Material properties
   and Light sources

4. The rendering pipeline

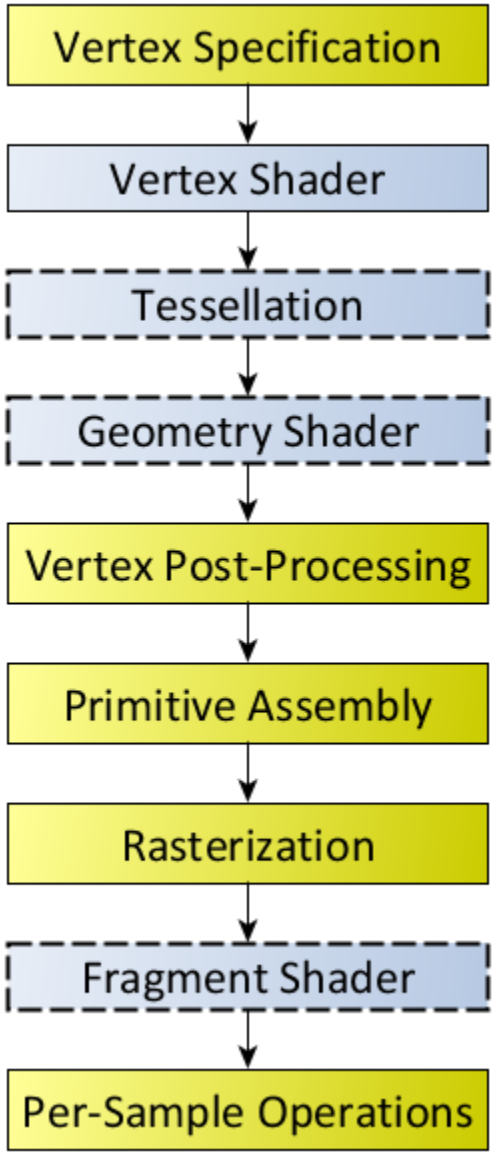


4. The rendering pipeline

Pràctica 1. El pipeline clàssic
El meu primer *vertex shader*
Introducció del *fragment shader*.
Pintar un objecte i col·locar una càmera.
Phong rendering model, Toon rendering

Pràctica 2. Geometria
Generació de geometria en la targeta gràfica. El meu primer *geometry shader* (per a fer aquesta pràctica caldrà haver completat la primera pràctica de Mecànica)

Pràctica 3. OpenGL avançat
Introducció de *frame buffer objects, stencil buffer, depth buffer.*
Optimització de computació gràfica en OpenGL 4

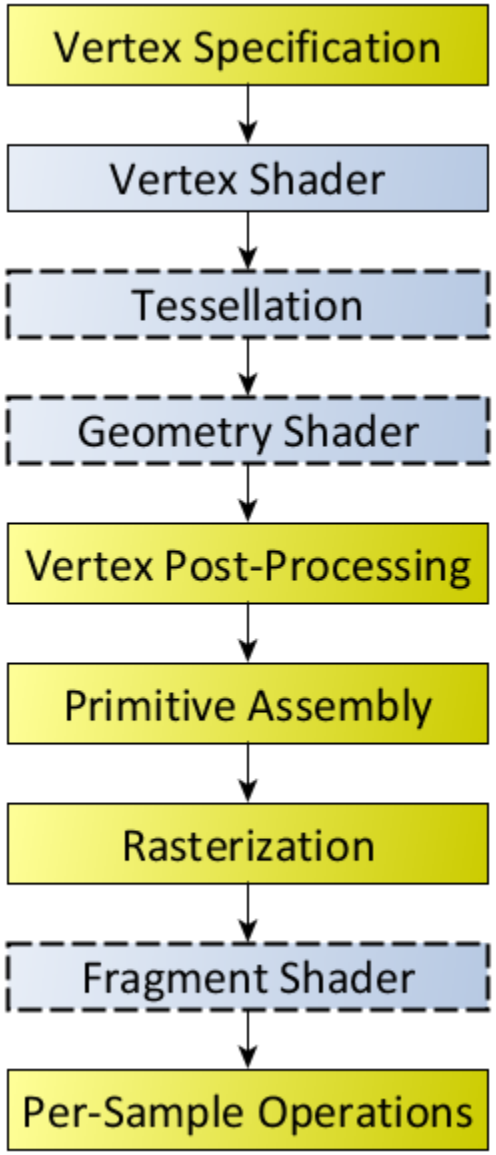


4. The rendering pipeline

Pràctica 1. El pipeline clàssic
El meu primer *vertex shader.*
Introducció del *fragment shader*.
Pintar un objecte i col·locar una càmera.
Phong rendering model, Toon rendering

Pràctica 2. Geometria
Generació de geometria en la targeta gràfica. El meu primer *geometry shader* (per a fer aquesta pràctica caldrà haver completat la primera pràctica de Mecànica)

Pràctica 3. OpenGL avançat
Introducció de *frame buffer objects, stencil buffer, depth buffer.*
Optimització de computació gràfica en OpenGL 4

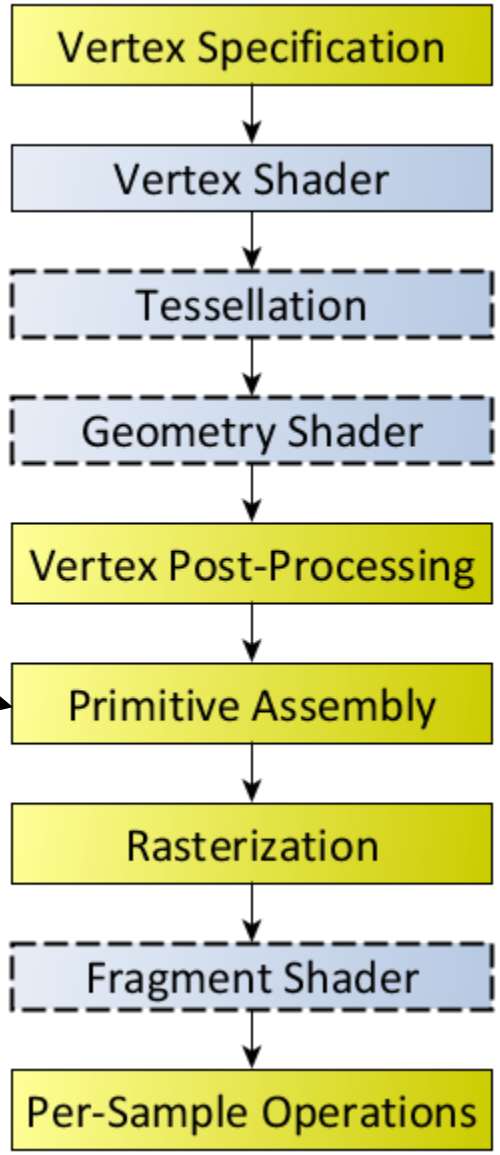


4. The rendering pipeline

Pràctica 1. El pipeline clàssic
El meu primer *vertex shader.*
Introducció del *fragment shader*.
Pintar un objecte i col·locar una càmera.
Phong rendering model, Toon rendering

Pràctica 2. Geometria
Generació de geometria en la targeta gràfica. El meu primer *geometry shader* (per a fer aquesta pràctica caldrà haver completat la primera pràctica de Mecànica)

Pràctica 3. OpenGL avançat
Introducció de *frame buffer objects, stencil buffer, depth buffer.*
Optimització de computació gràfica en OpenGL 4

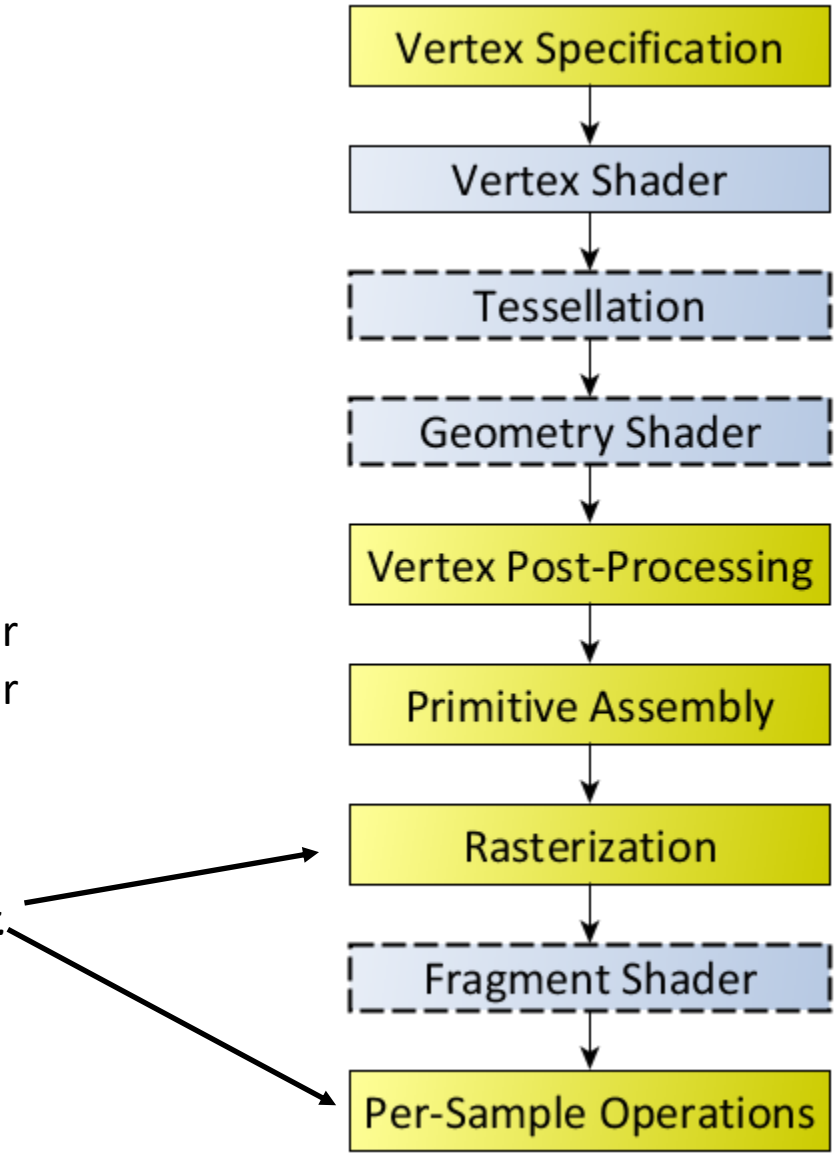


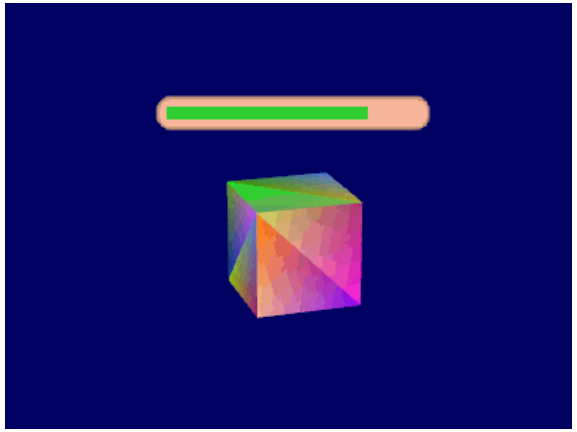
4. The rendering pipeline

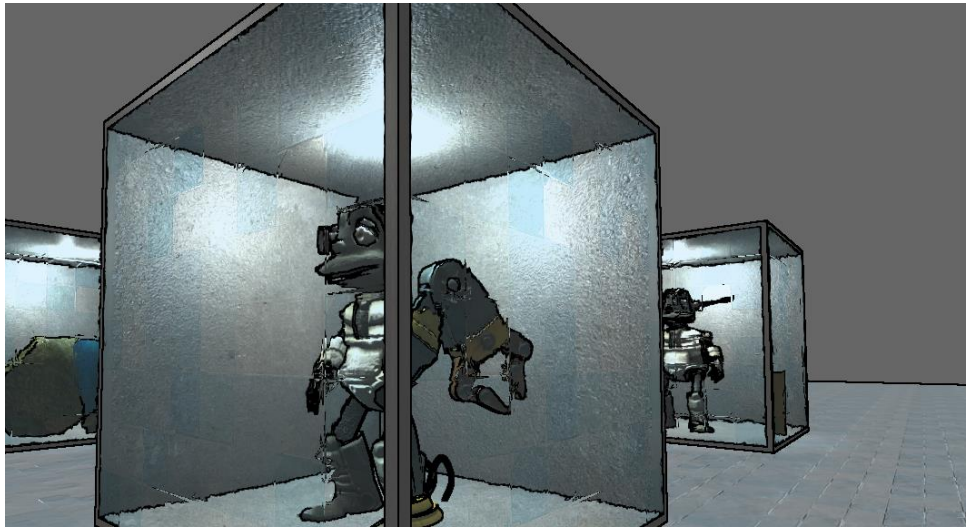
Pràctica 1. El pipeline clàssic
El meu primer *vertex shader.*
Introducció del *fragment shader*.
Pintar un objecte i col·locar una càmera.
Phong rendering model, Toon rendering

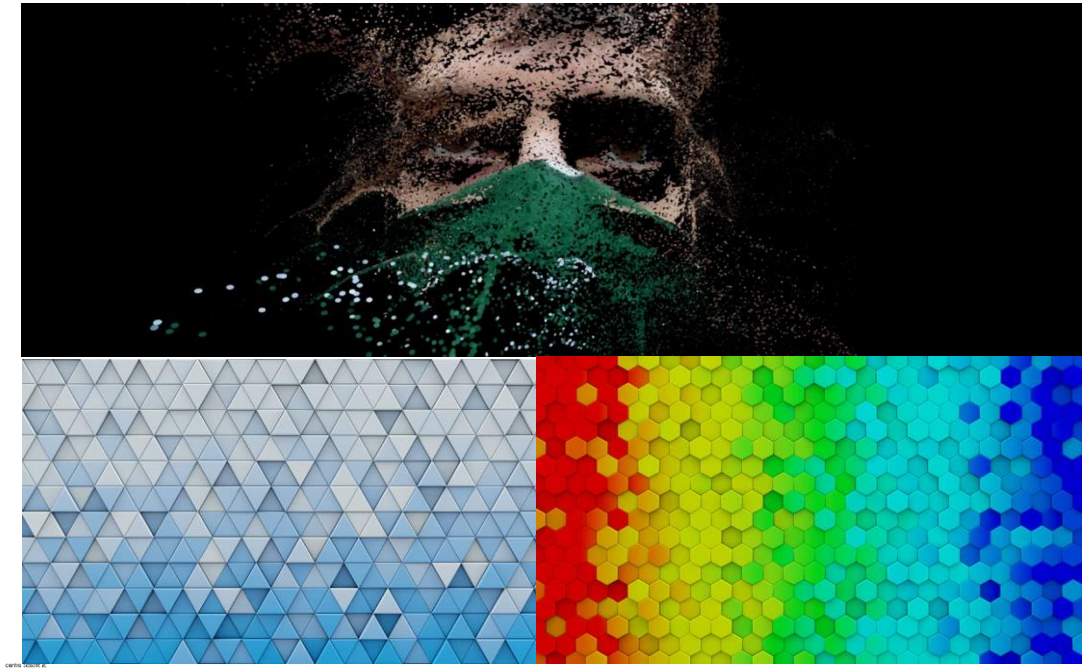
Pràctica 2. Geometria
Generació de geometria en la targeta gràfica. El meu primer *geometry shader* (per a fer aquesta pràctica caldrà haver completat la primera pràctica de Mecànica)

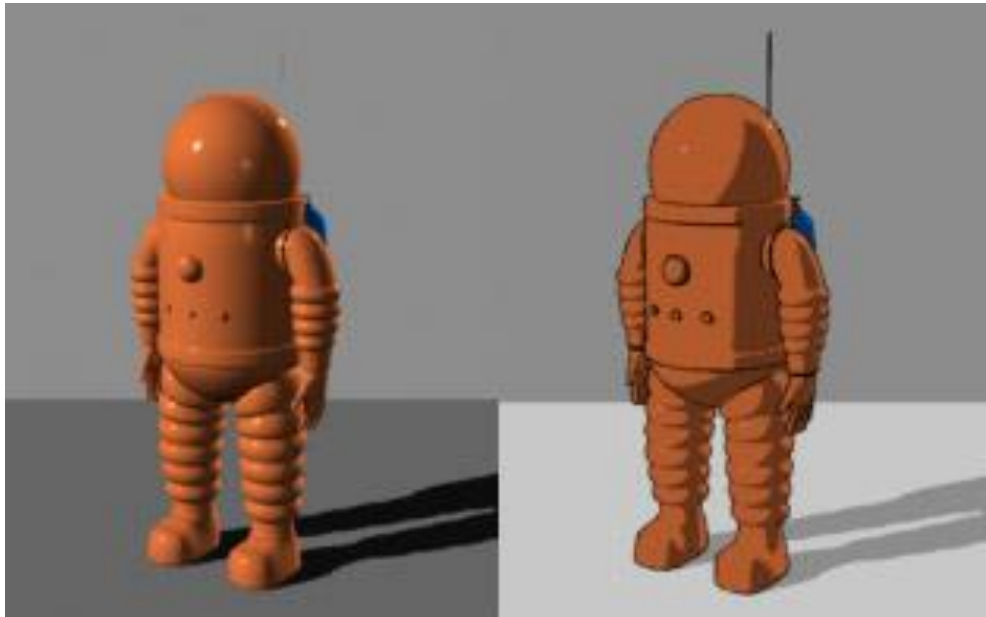
Pràctica 3. OpenGL avançat
Introducció de *frame buffer objects, stencil buffer, depth buffer.*
Optimització de computació gràfica en OpenGL 4











Resources

- [Angel2011] Edward Angel, Dave Shreiner (2011) *Interactive Computer Graphics: A Top-down Approach Using Opengl*, 6th Edition. Pearson education

- [Khronos,2018] https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview, accessed 02/2018