

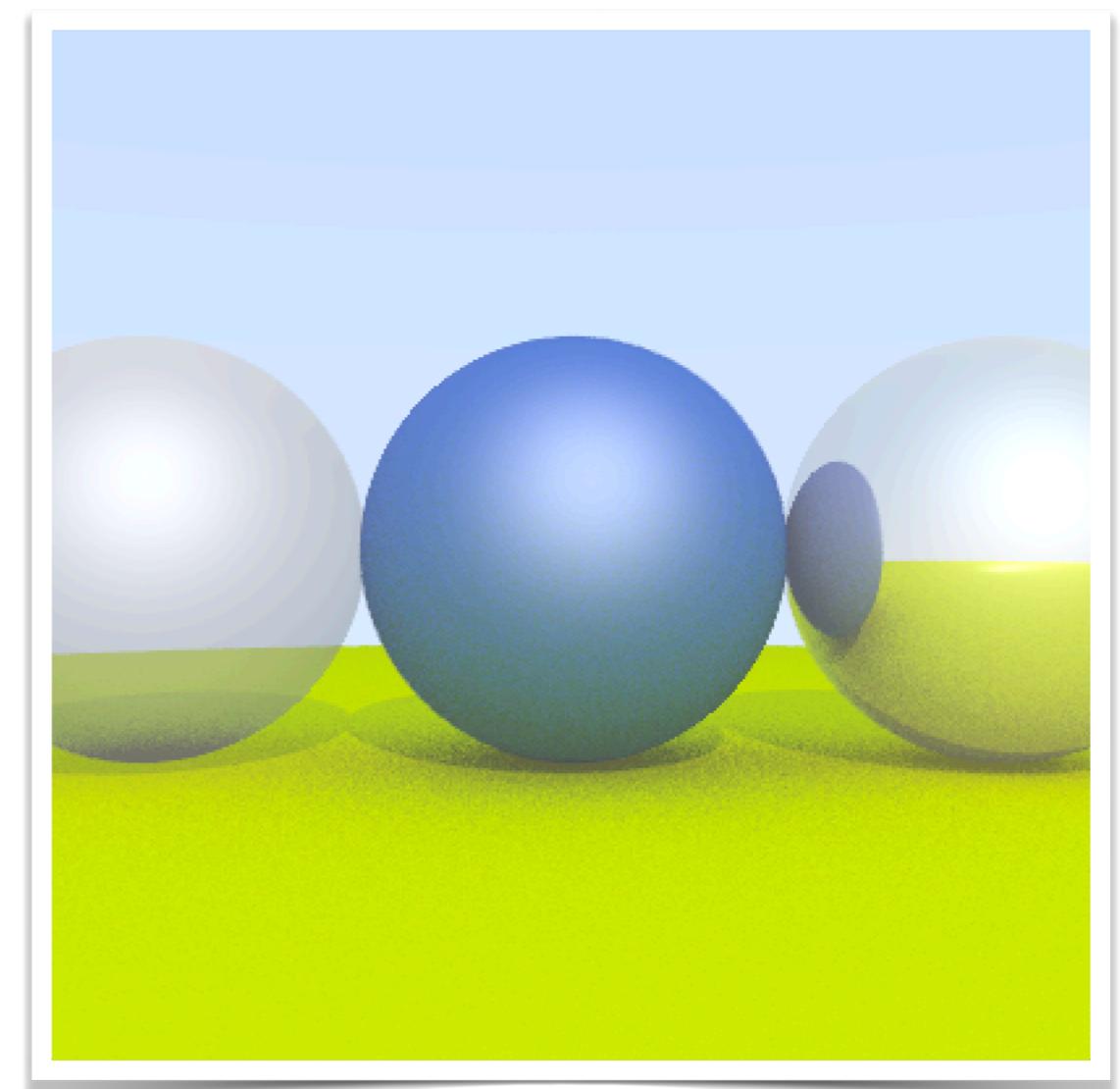
# **Sessió Setmana 3**

**GiVD 2022-23**

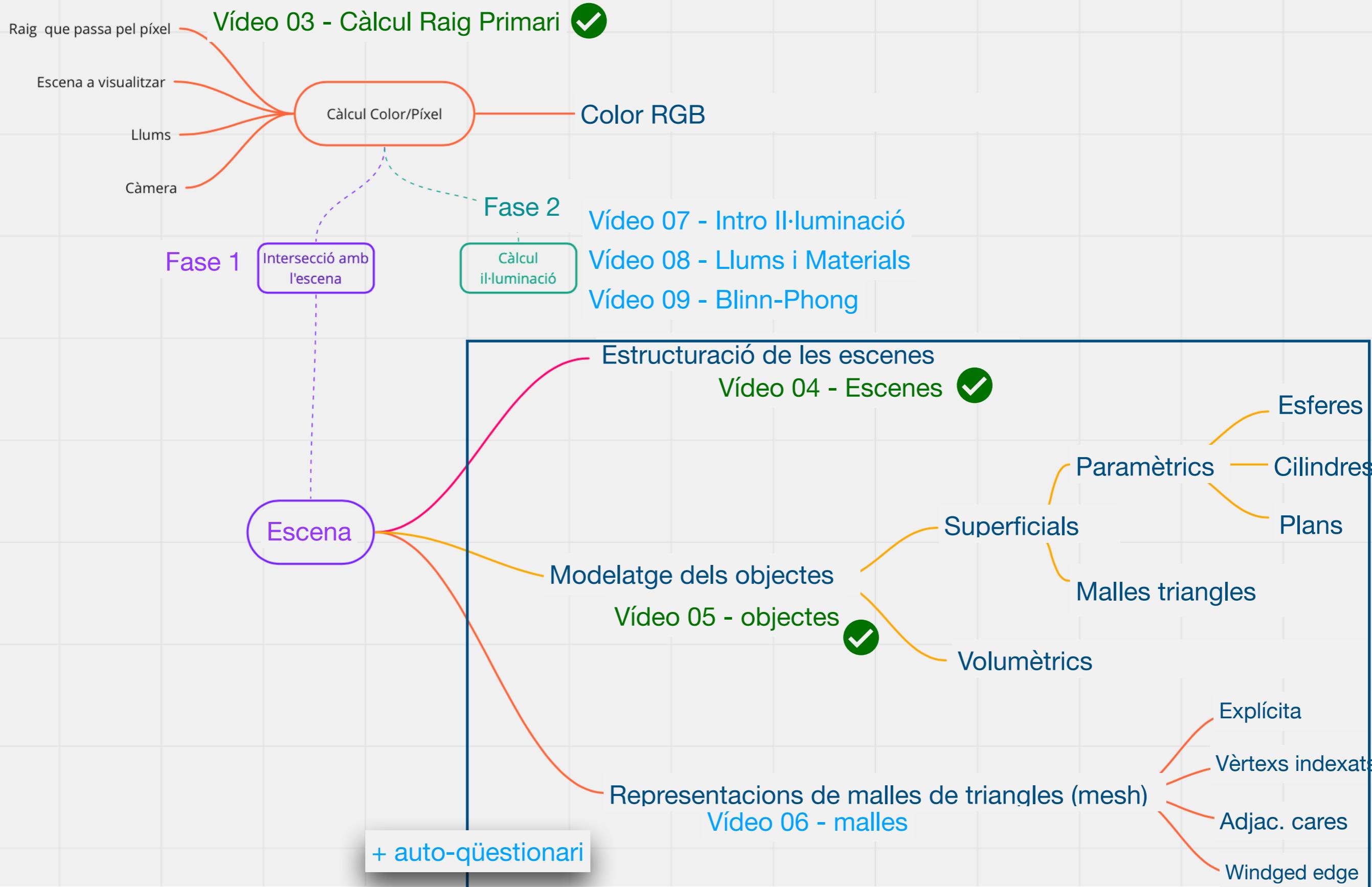
# Guió de la sessió

---

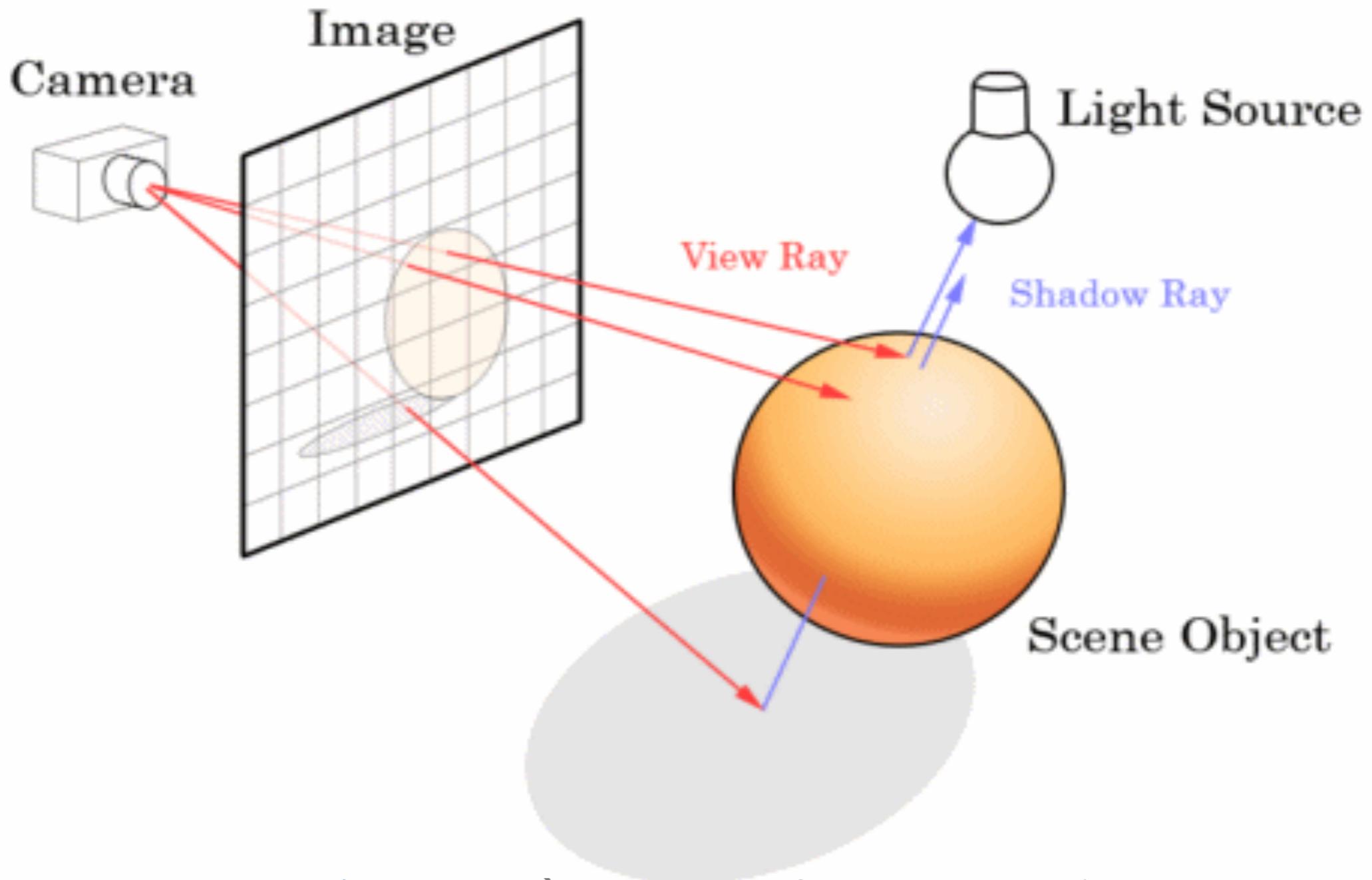
1. Planificació de la setmana 3
  2. Raig i càmera
  3. Activitat
  4. Interseccions: escenes, objectes, malles
  5. Comentaris i dubtes de la pràctica
- 



# 1. Planificació



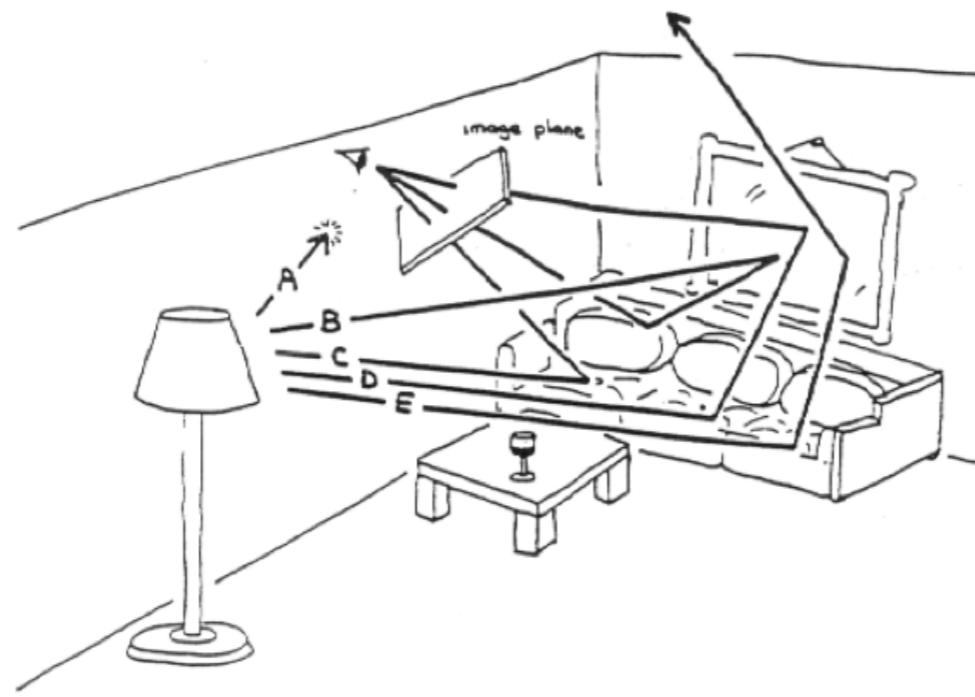
## 2. Raig i càmera



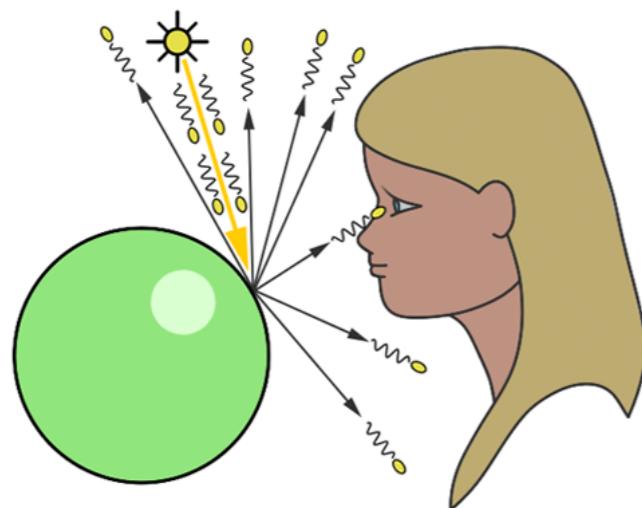
- **Problema a solucionar:** càcul dels fotons que impacten a la imatge

# 2. Raig i càmera

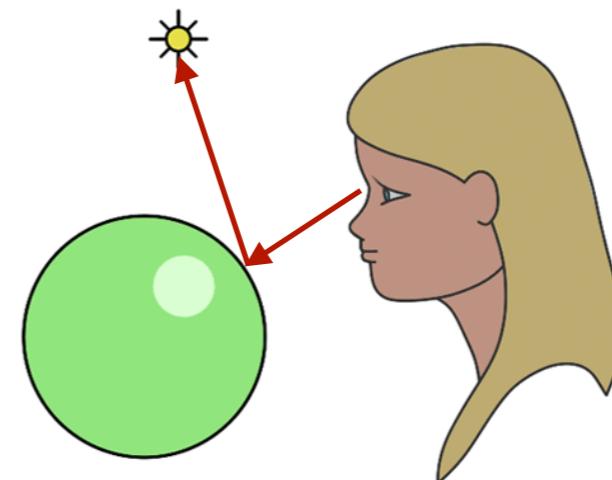
- **Problema a solucionar:** càcul dels fotons que impacten a la imatge.



**Forward RayTracing (photon tracing)**

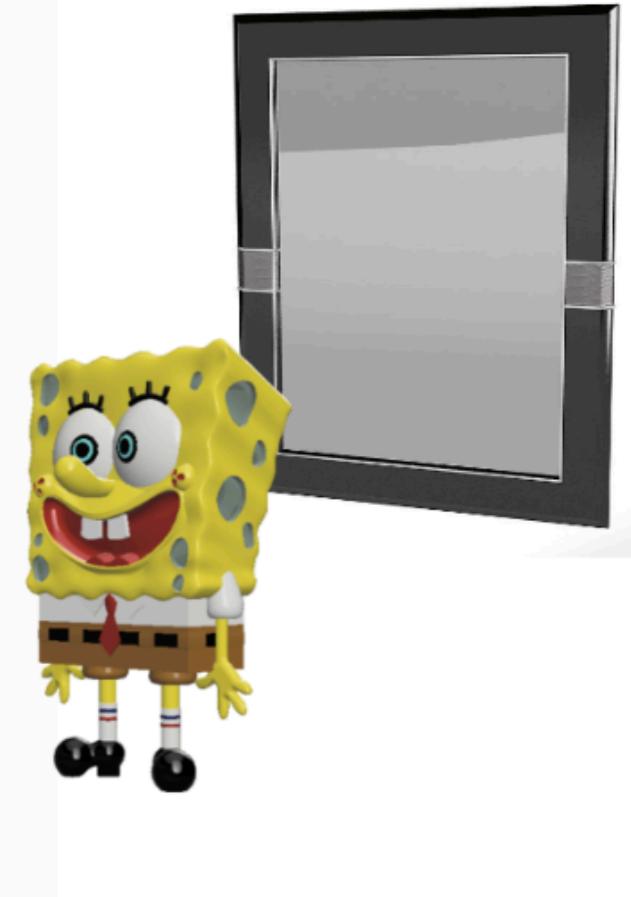
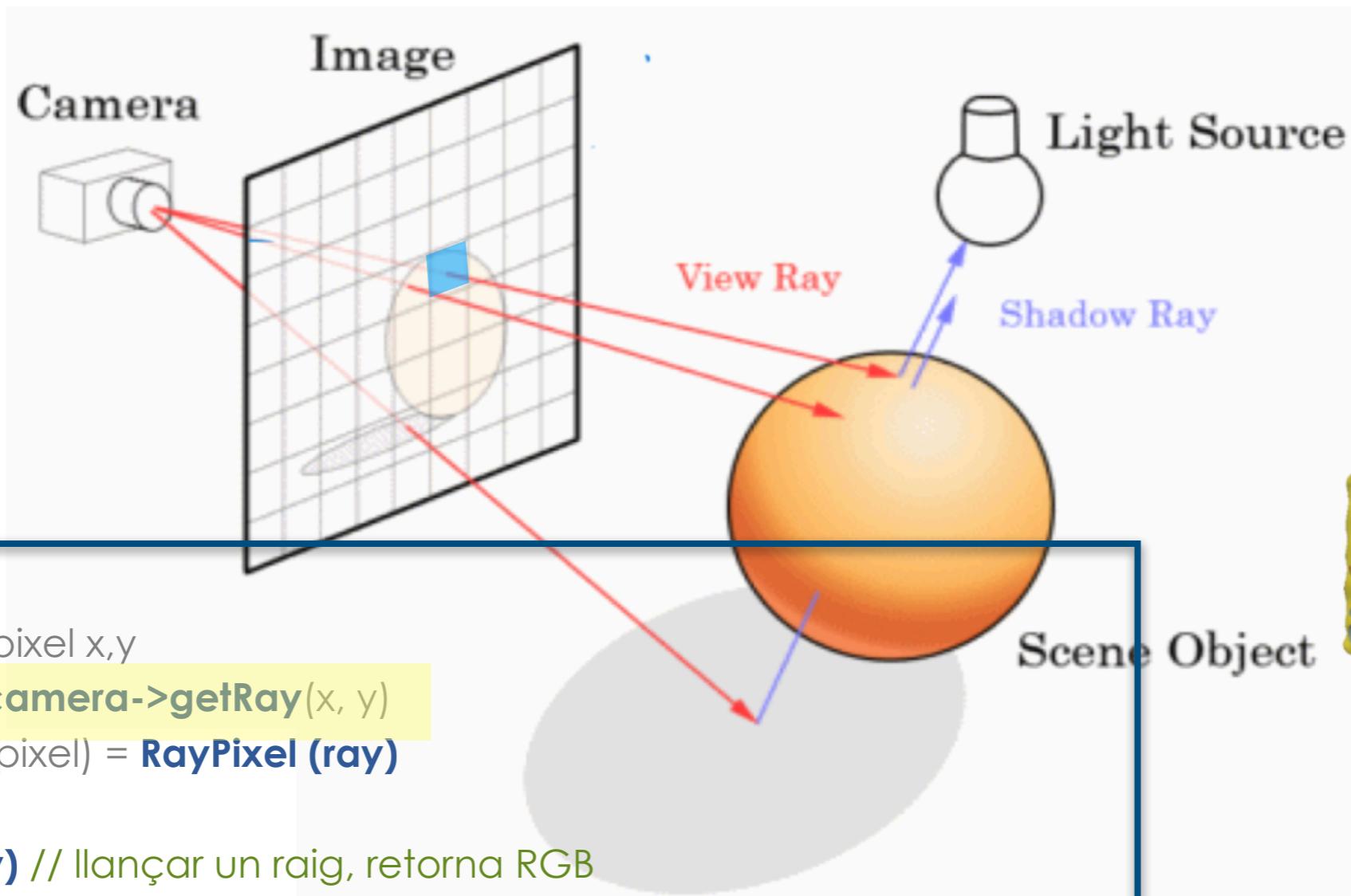


**Backward RayTracing**



<https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/raytracing-algorithm-in-a-nutshell.html>

# 2. Raig i Càmera

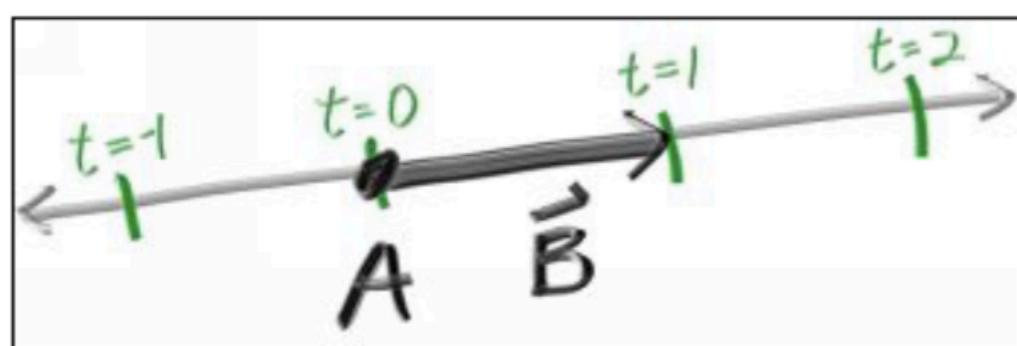
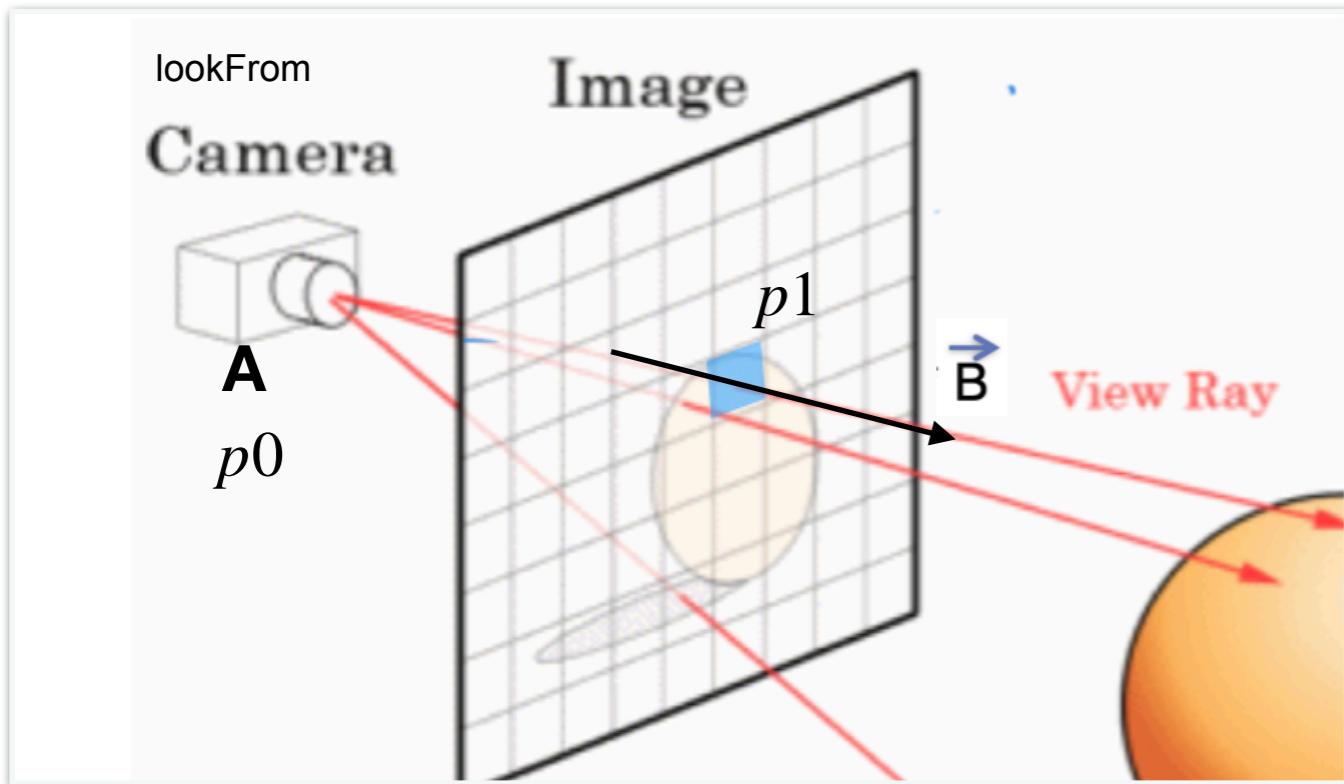
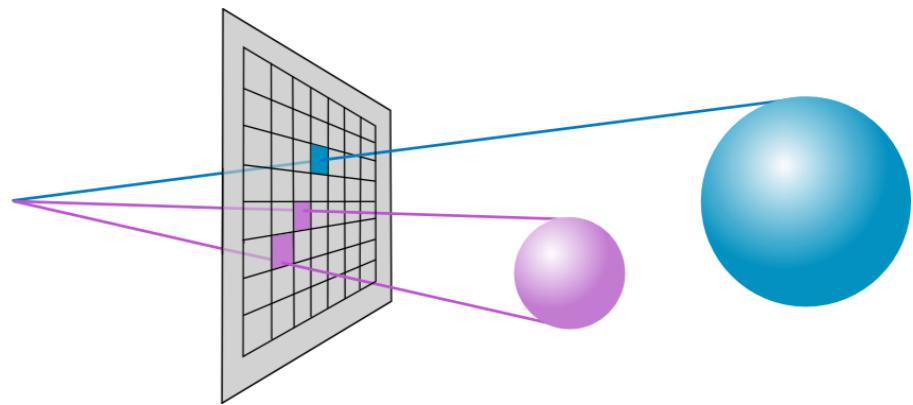


**Classe Raytracer.cpp**

**Classe Scene.cpp**

- **hit(ray)**  
per cada objecte o en l'escena  
    o->**hit(ray, surface)** //hit(ray, tmin, tmax, hitinfo)  
retorna el punt més proper a l'observador (+normal, +material)
- **Shade(hitInfo, ray)** // retorna la il·luminació en el point  
retorna color

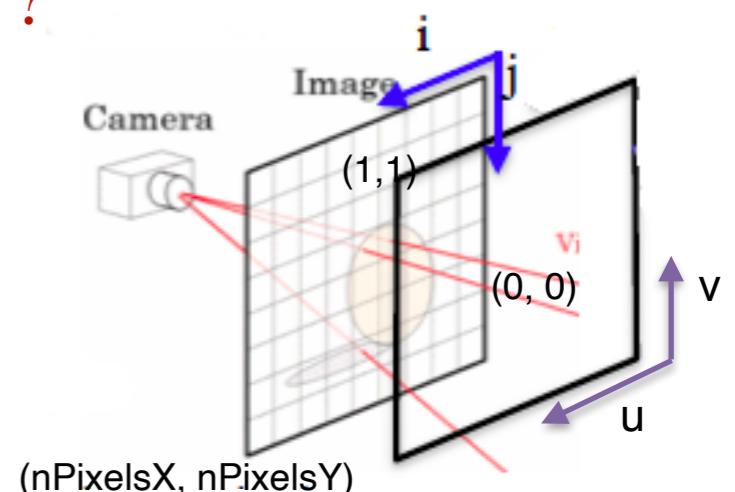
# ComputeRay/getRay



$$\vec{B} = \text{glm} :: \text{normalize}(p_1 - p_0)$$

$$p_0 = A$$

$$p_1 = ??$$



$$p = A + t * \vec{B}$$

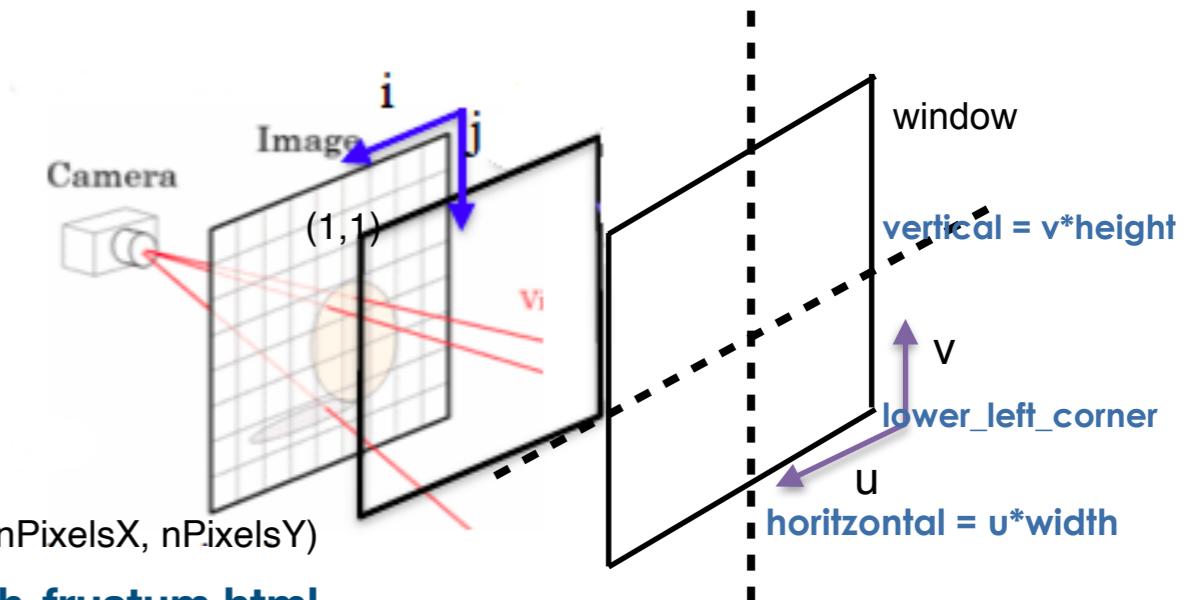
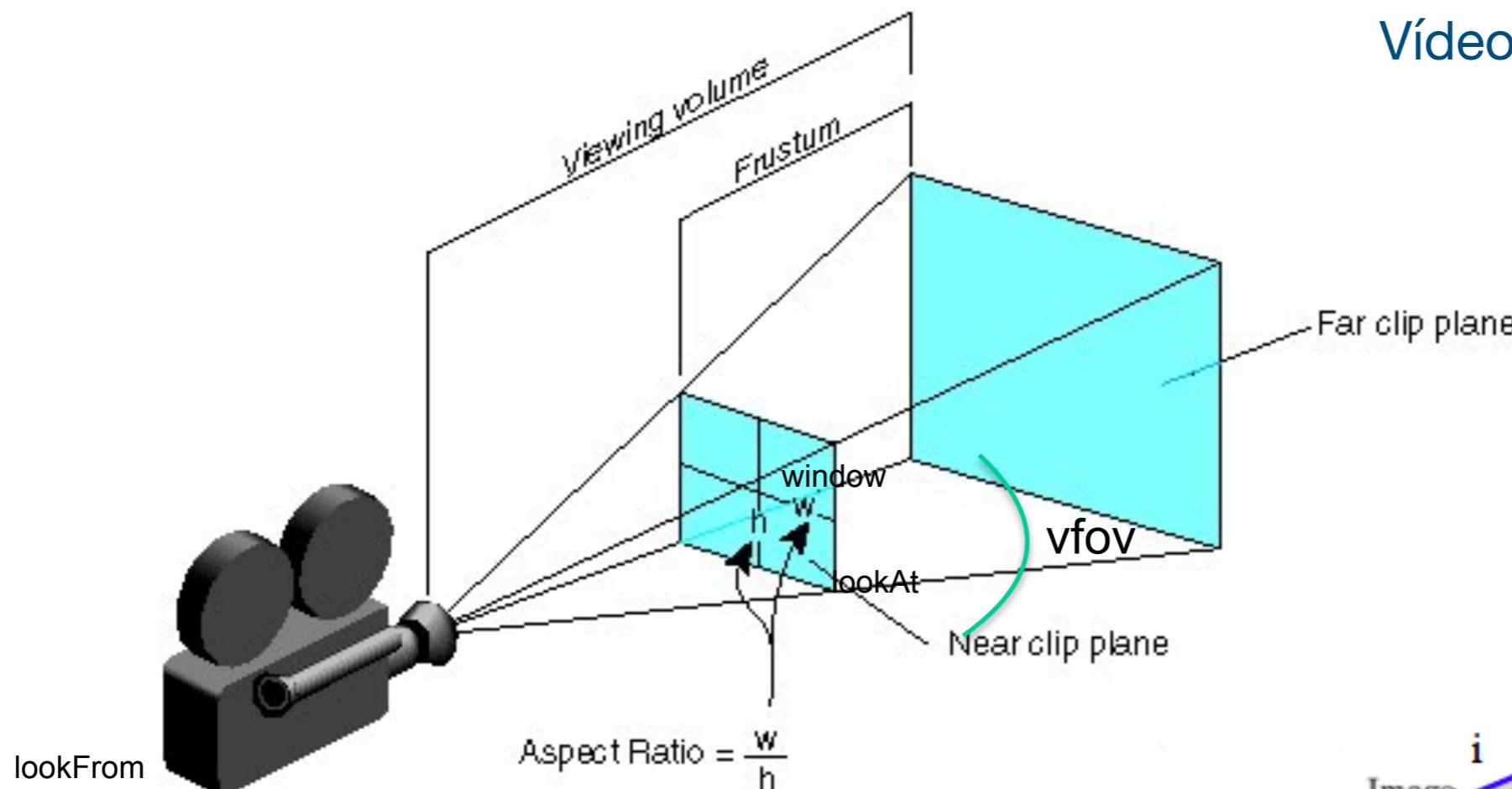
A punt inicial

B vector director

t paràmetre de la recta

# ComputeRay/getRay

Vídeo 03 - Càlcul Raig Primari

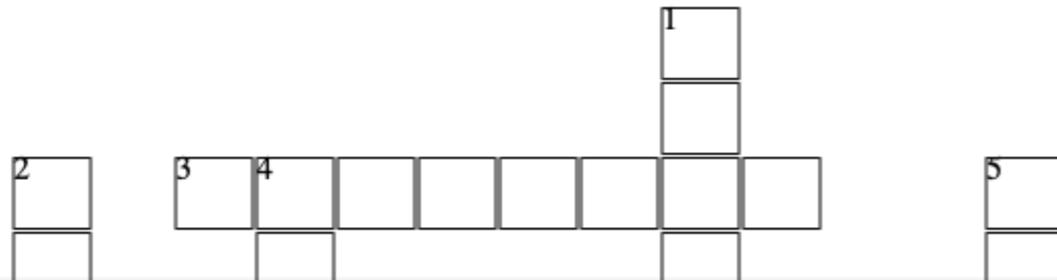


<https://webglfundamentals.org/webgl/webgl-visualize-camera-with-frustum.html>

<https://cs.wellesley.edu/~cs307/threejs/demos/Camera/camera-api.shtml>

$$lower\_left\_corner + horizontal * u_i + vertical * v_i$$

# 2. Raig i Càmera

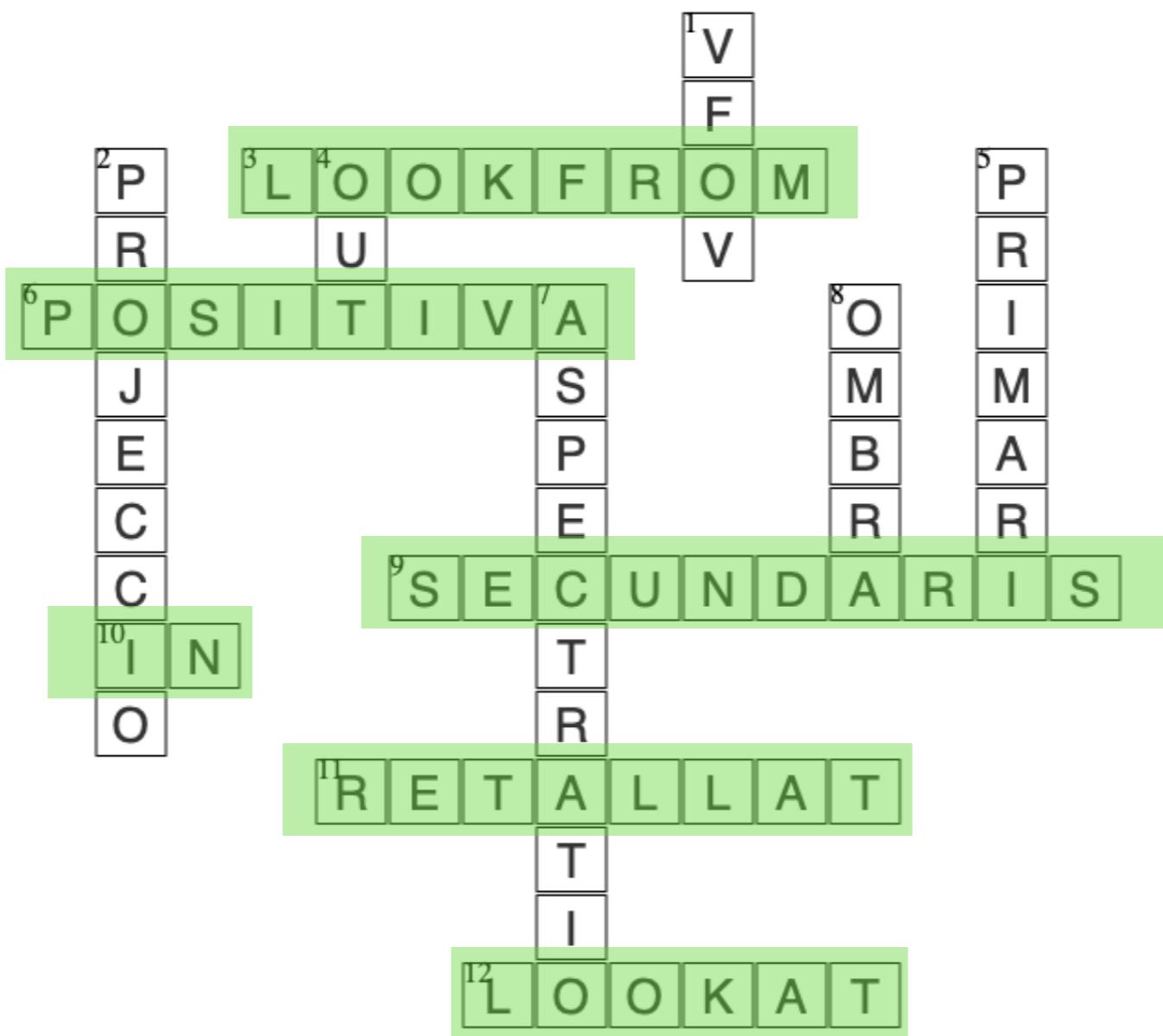


## Horizontales

- 3 Punt on comença el raig primari (en anglès)
- 6 Per tenir en compte una intersecció amb qualsevol raig, sempre serà...
- 9 Rajos de reflexió i de transmissió
- 10 Si decremento el vfov, faré un zoom...
- 11 Plans de .... (limiten la part visible de l'escena)
- 12 Punt on s'enfoca (en anglès)

## Verticales

- 1 Angle d'obertura vertical de la càmera
- 2 La finestra (o window) es situa en el pla de ....
- 4 Si augmento la mida de la meva finestra o window, faré un zoom ...
- 5 S'anomena així al raig que va des de l'observador a un píxel.
- 7 Relació que permet fixar la mida de la finestra amb ajuda de vfov (dues paraules en anglès)
- 8 S'anomena així al raig que va d'un punt a la llum

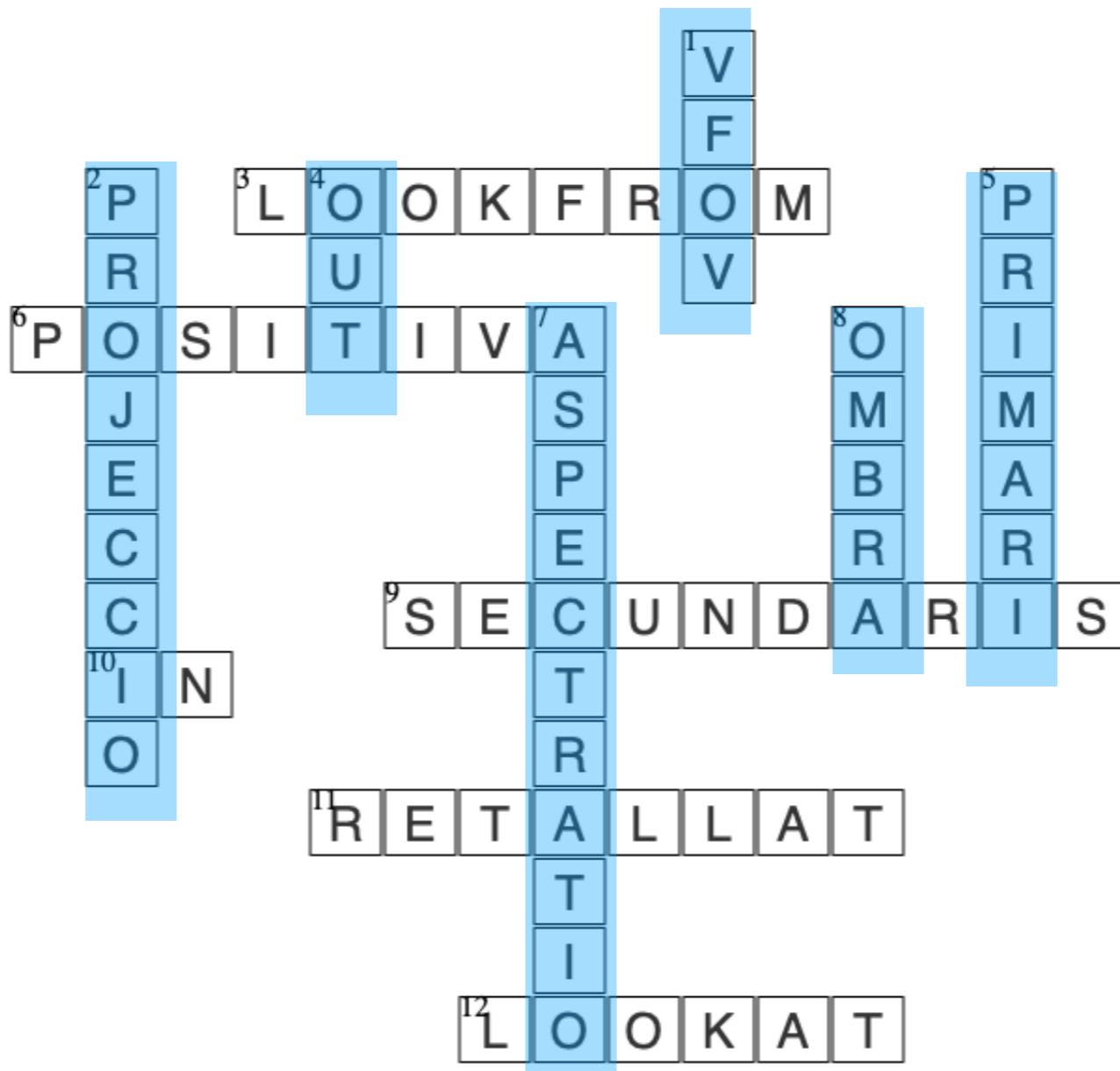


## Horizontals

- 3 Punt on comença el raig primari (en anglès)
- 6 Per tenir en compte una intersecció amb qualsevol raig, sempre serà...
- 9 Rajos de reflexió i de transmissió
- 10 Si decremento el vfov, faré un zoom...
- 11 Plans de .... (limiten la part visible de l'escena)
- 12 Punt on s'enfoca (en anglès)

## Verticales

- 1 Angle d'obertura vertical de la càmera
- 2 La finestra (o window) es situa en el pla de ....
- 4 Si augmento la mida de la meva finestra o window, faré un zoom ...
- 5 S'anomena així al raig que va des de l'observador a un píxel.
- 7 Relació que permet fixar la mida de la finestra amb ajuda de vfov (dues paraules en anglès)
- 8 S'anomena així al raig que va d'un punt a la llum



## Horizontales

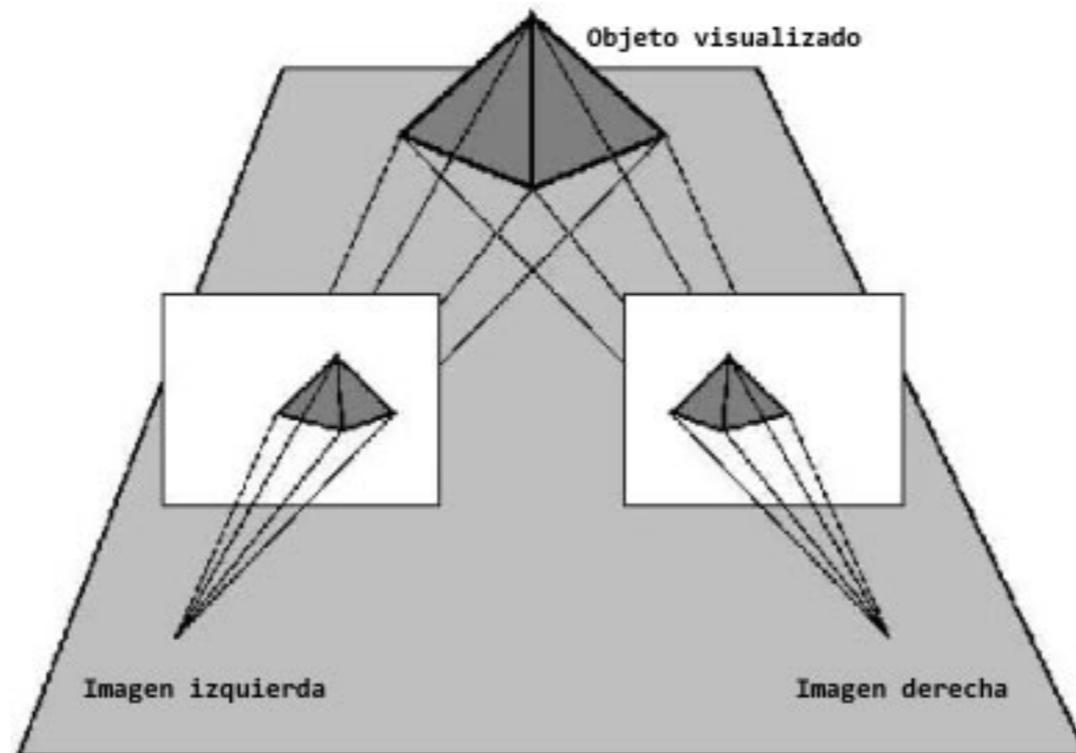
- 3 Punt on comença el raig primari (en anglès)
- 6 Per tenir en compte una intersecció amb qualsevol raig, sempre serà...
- 9 Rajos de reflexió i de transmissió
- 10 Si decremento el vfov, faré un zoom...
- 11 Plans de .... (limiten la part visible de l'escena)
- 12 Punt on s'enfoca (en anglès)

## Verticales

- 1 Angle d'obertura vertical de la càmera
- 2 La finestra (o window) es situa en el pla de ....
- 4 Si augmento la mida de la meva finestra o window, faré un zoom ...
- 5 S'anomena així al raig que va des de l'observador a un píxel.
- 7 Relació que permet fixar la mida de la finestra amb ajuda de vfov (dues paraules en anglès)
- 8 S'anomena així al raig que va d'un punt a la llum

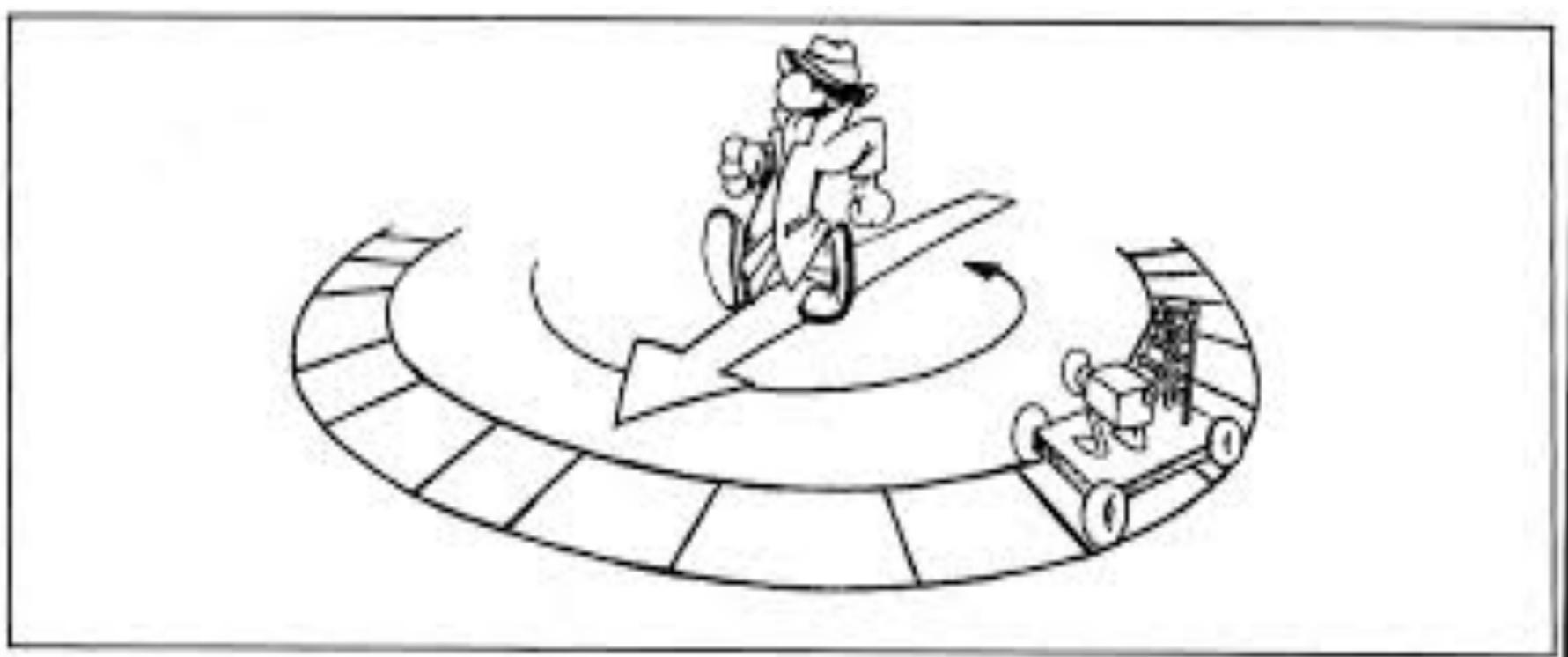
# Kahoot!

- Per a obtenir una visió esteroscòpica, es construeixen dues imatges (una per a cada ull) col·locant l'observador en dues posicions molt properes. Si es vol visualitzar tota l'escena i es considera un observador al punt  $(x, y, z)$ , però que la separació entre els seus ulls és  $\Delta x$ , quins atributs de les dues càmeres hauran de ser diferents?



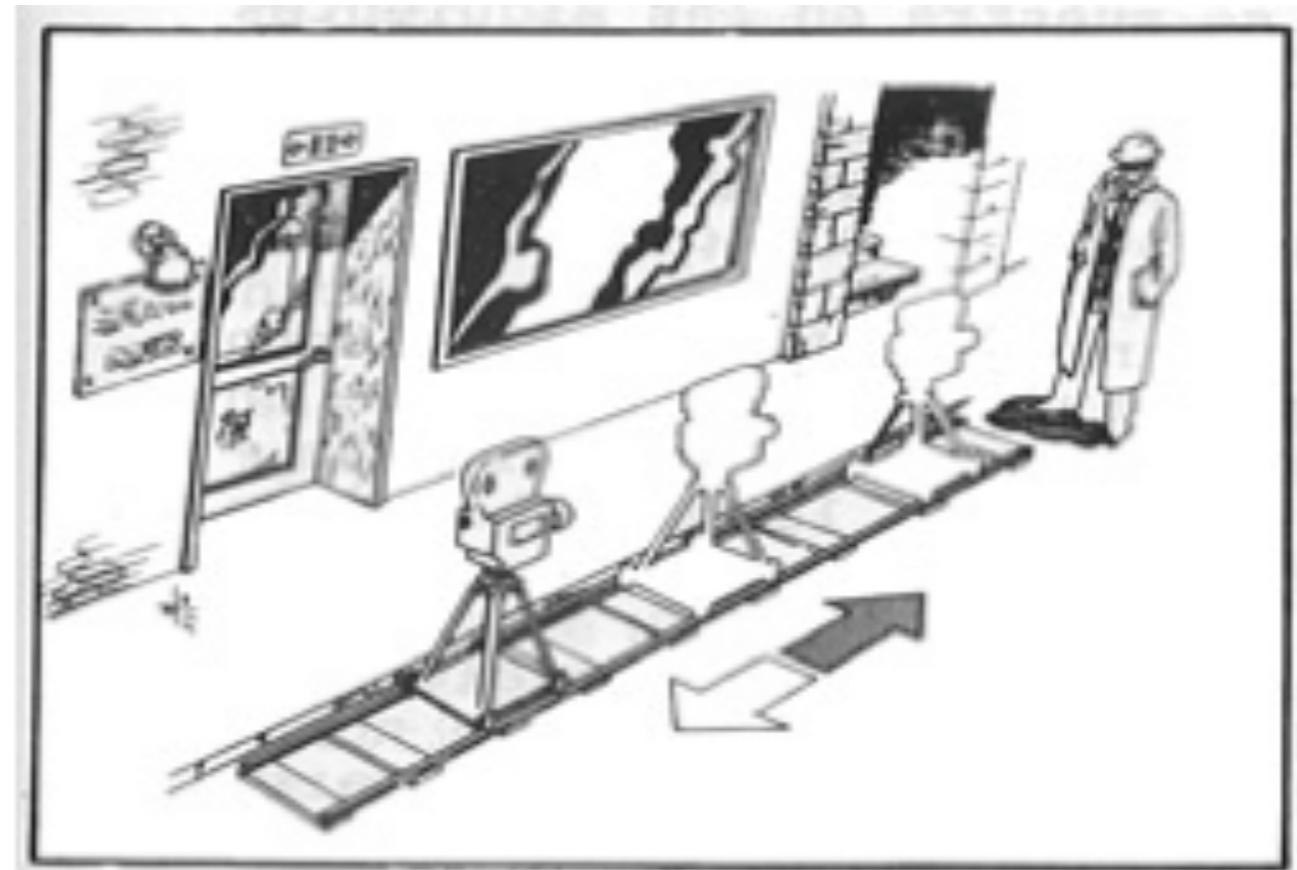
# Kahoot!

- En alguns jocs, es vol fer un moviment de la càmera automàtic, de forma que el personatge és el centre d'interès i la càmera fa voltes circulars al seu voltant per a gravar d'on venen els projectils. Quins paràmetres de la càmera canviarien?



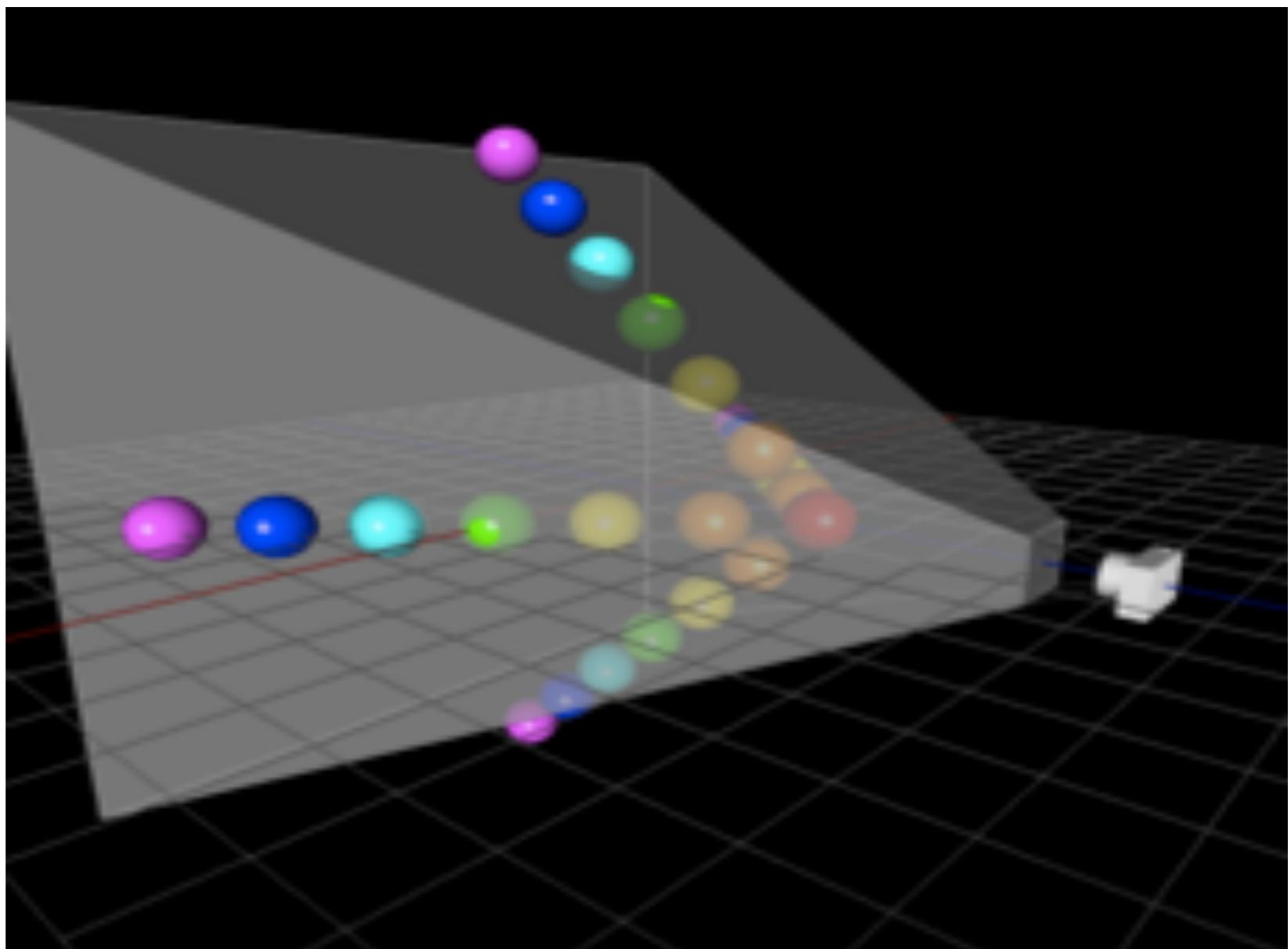
# Kahoot!

- En alguns jocs, es vol fer un moviment de la càmera perspectiva automàtic, de forma que el personatge és el centre d'interès i la càmera s'aproxima gradualment al personatge, fent un moviment de *travelling*, com indica la figura. Quins paràmetres de la càmera cal canviar?
  - A) Si el personatge no es mou
  - B) Si el personatge es mou



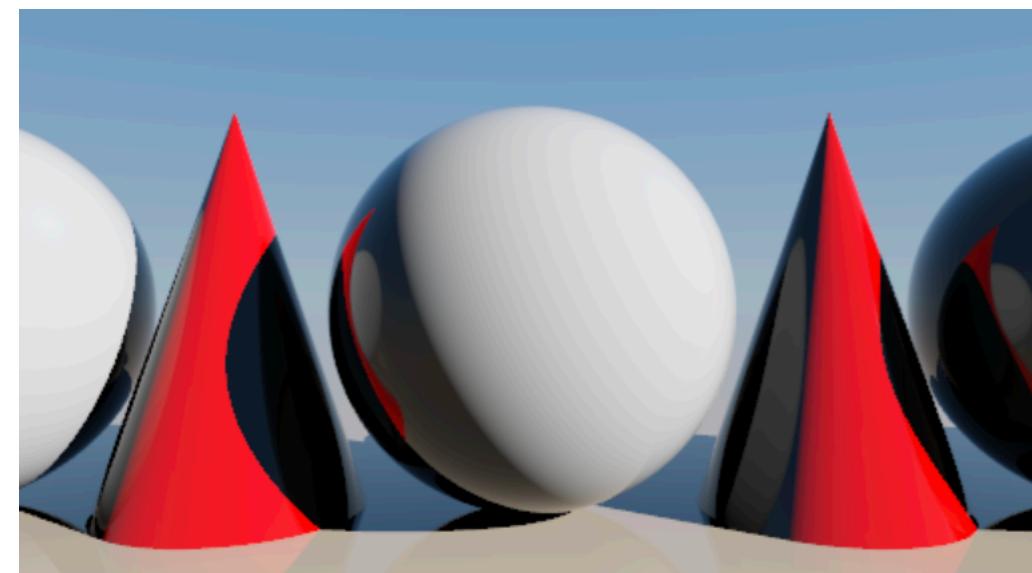
# Kahoot!

- Què es faria per fer zoom de l'escena? Quin paràmetre de la càmera s'ha de modificar?



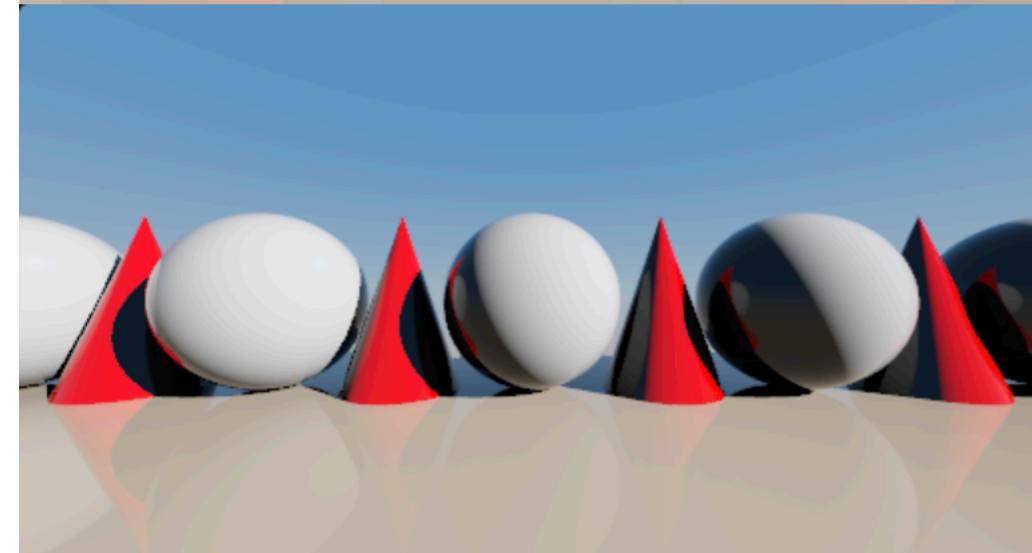
# Kahoot!

**vfov=40**

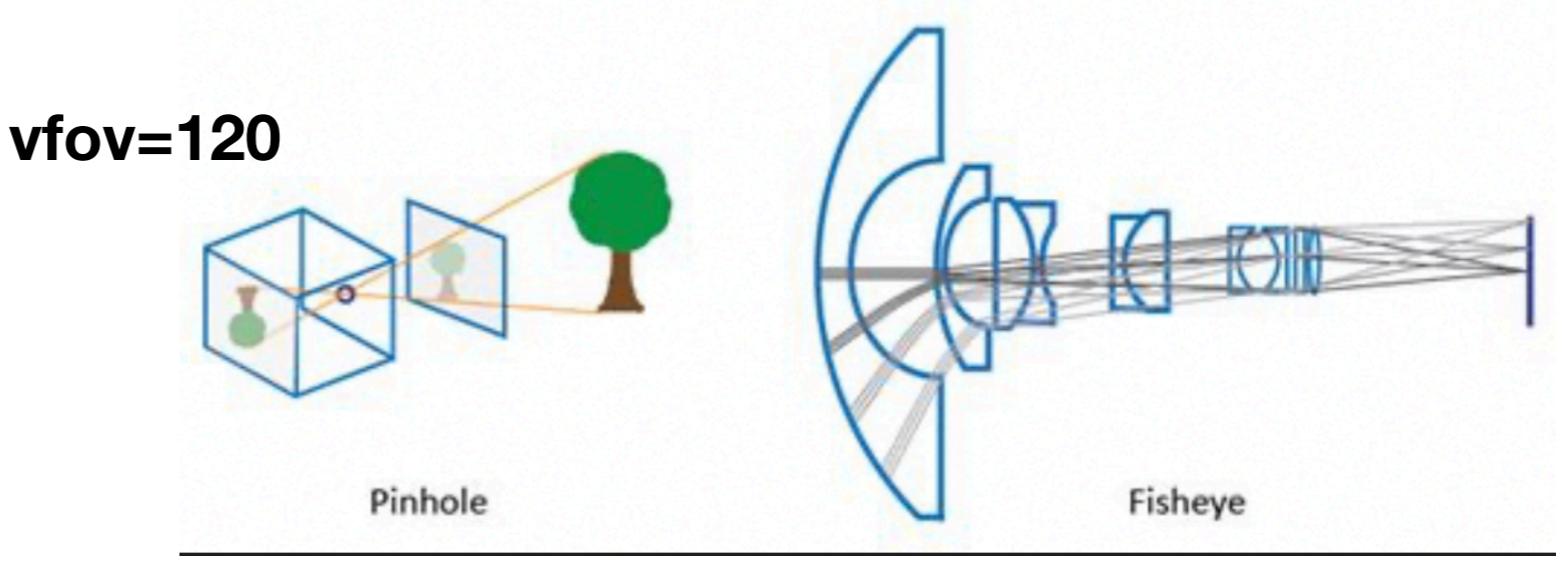


**vfov=80**

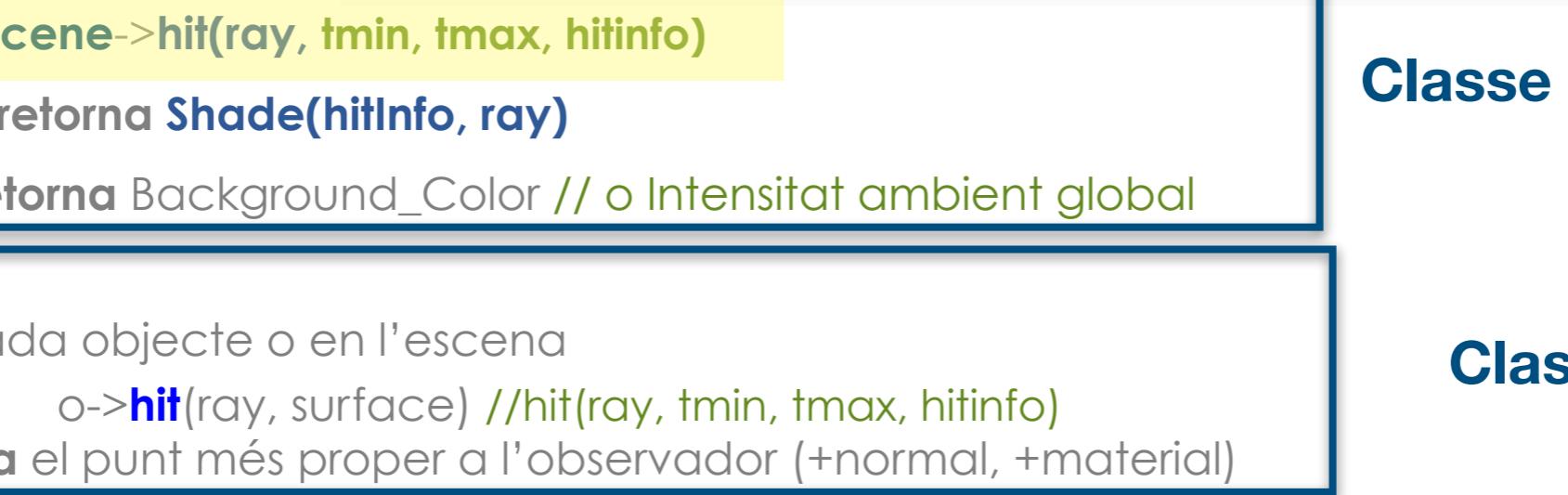
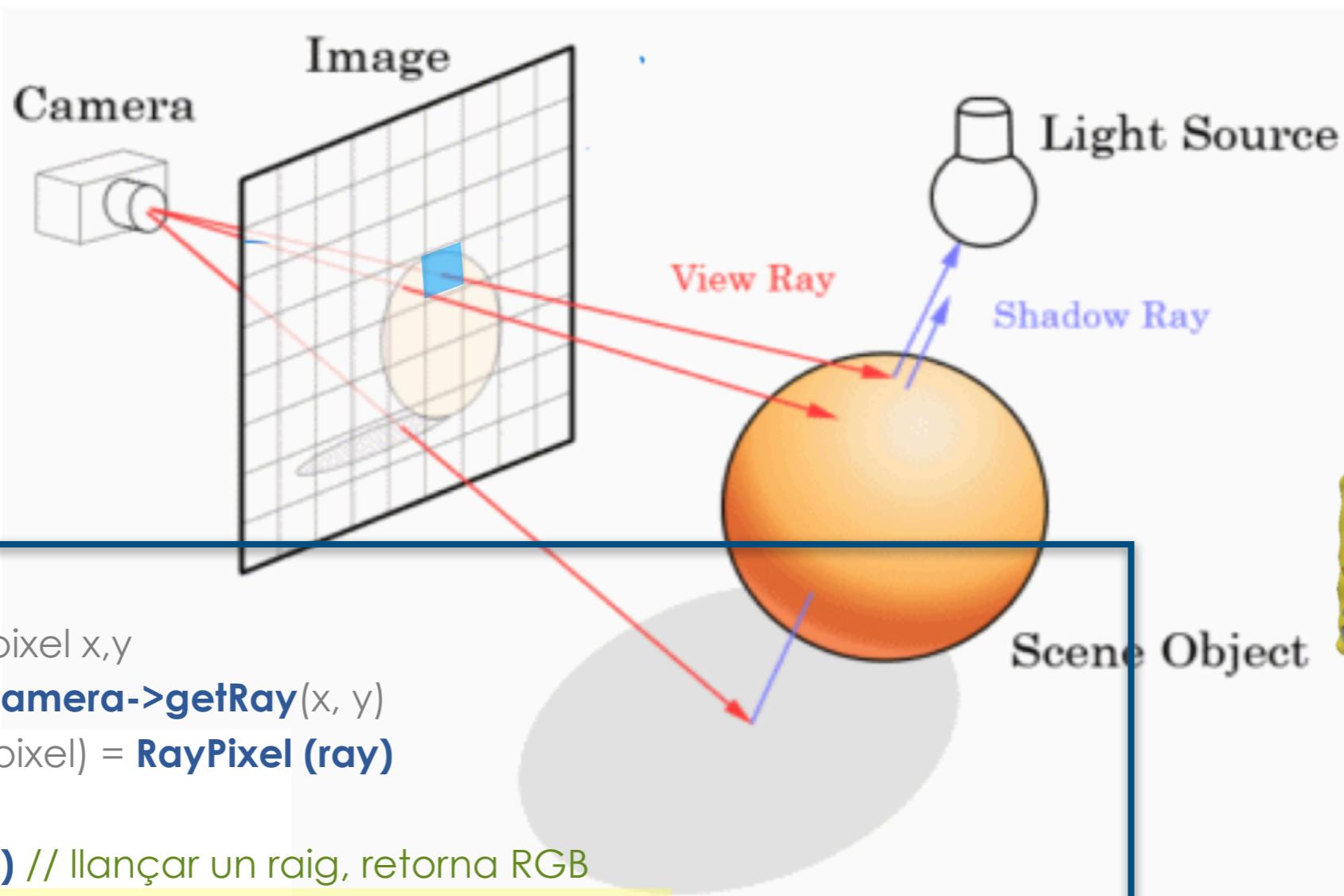
<https://www.shadertoy.com/view/MLtBW2>



**vfov=120**



# 4. Interseccions



**Classe Raytracer.cpp**

**Classe Scene.cpp**

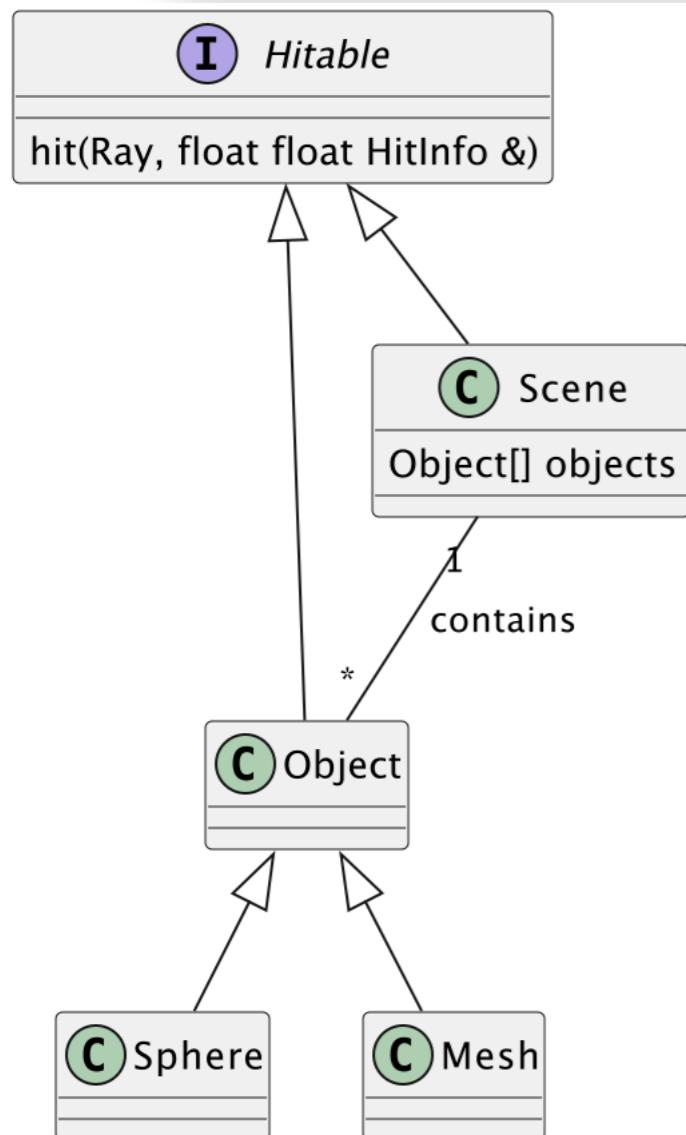
# 4. Interseccions

```

class Hitable
{
public:
    Hitable();
    ~Hitable();

    // Funció que retorna la intersecció més propera al t_min del raig. La intersecció
    // estarà entre t_min i t_max
    virtual bool hit(Ray& r, float tmin, float tmax, HitInfo& info) const = 0;

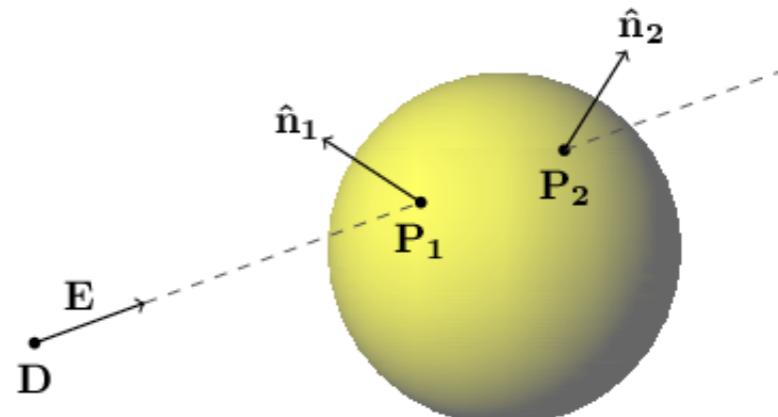
```



```

class HitInfo
{
public:
    float      t;          // t del raig on s'ha intersecat
    vec3       p;          // punt del raig on hi ha la intersecció
    vec3       normal;    // normal en el punt d'intersecció
    Material *mat_ptr;   // material de l'objecte que s'ha intersectat
    vec2       uv;         // punt 2D per la projecció de la textura

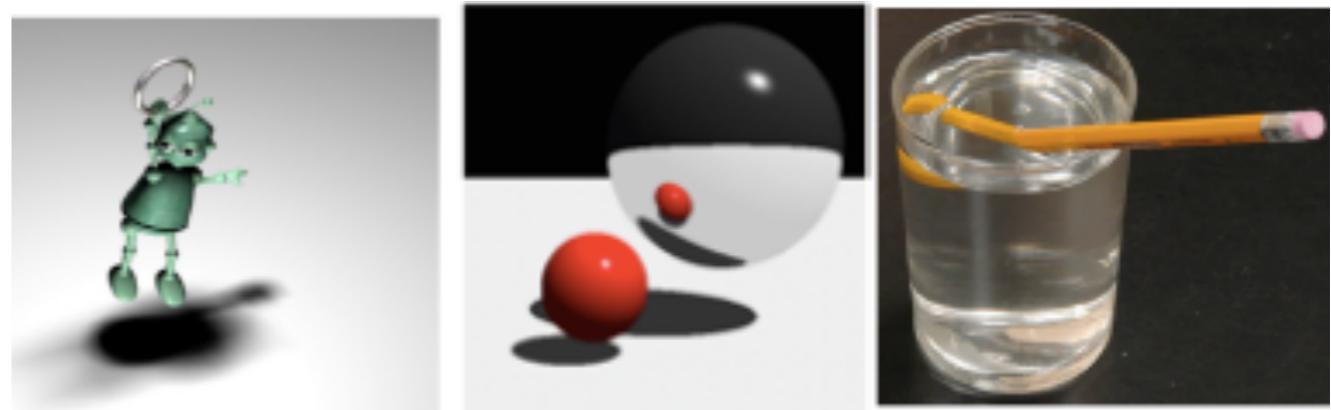
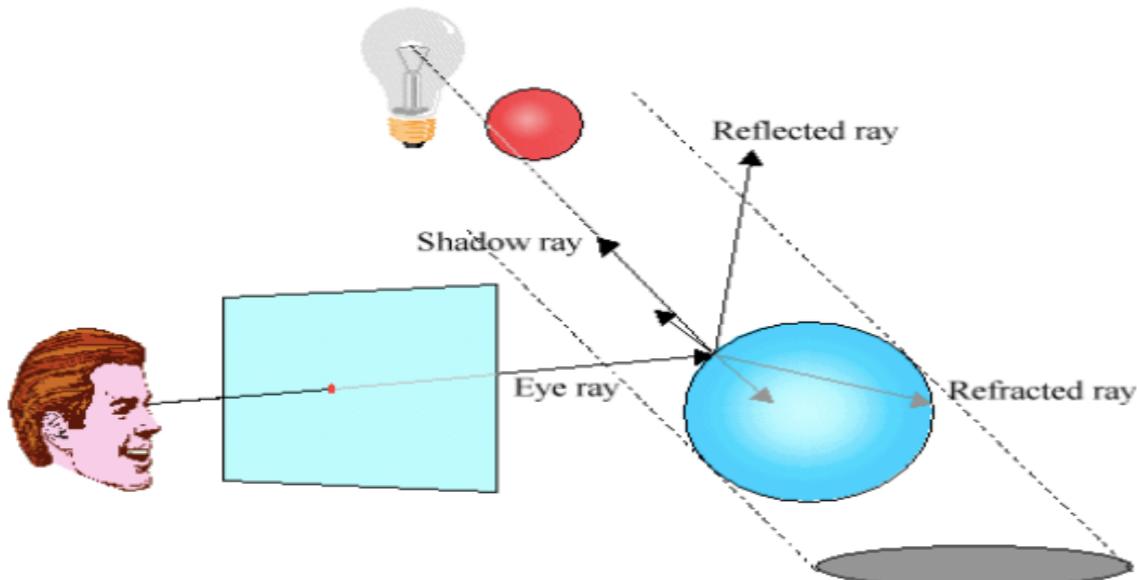
```



# Costos

## Algorisme recursiu: (RayPixel)

1. **Rajos primaris (eye ray)**: des de l'observador a l'escena
2. **Rajos d'ombra (shadow ray)**: des del punt de l'objecte a les llums
3. **Rajos secundaris**:
  - 3.1. **Rajos de reflexió (reflected ray)**: des del punt de l'objecte en direcció de mirall
  - 3.2. **Rajos de transparència (refracted ray)**: des del punt de l'objecte en direcció de refracció

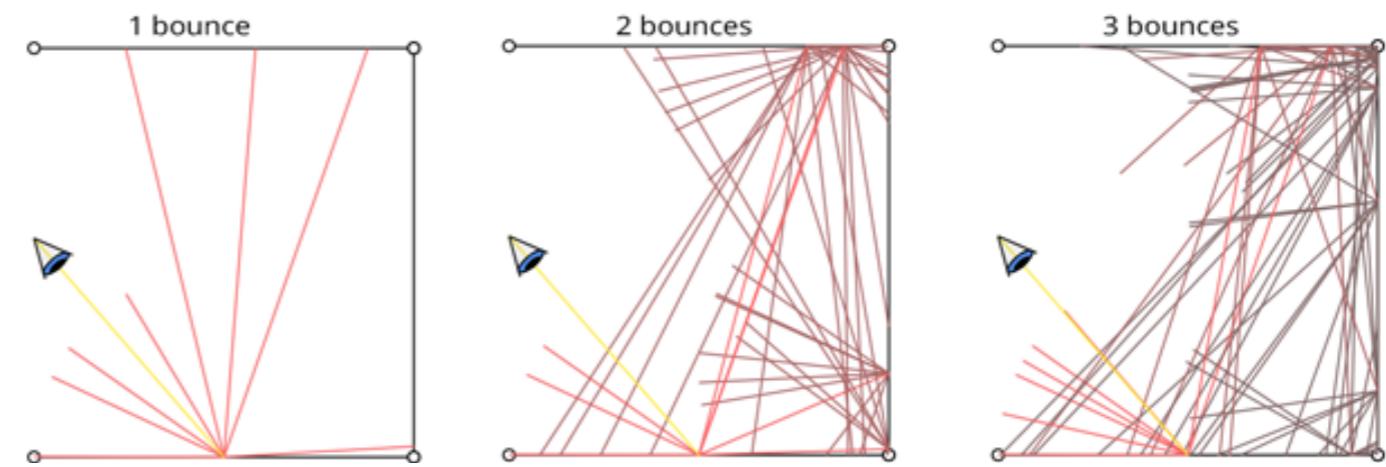
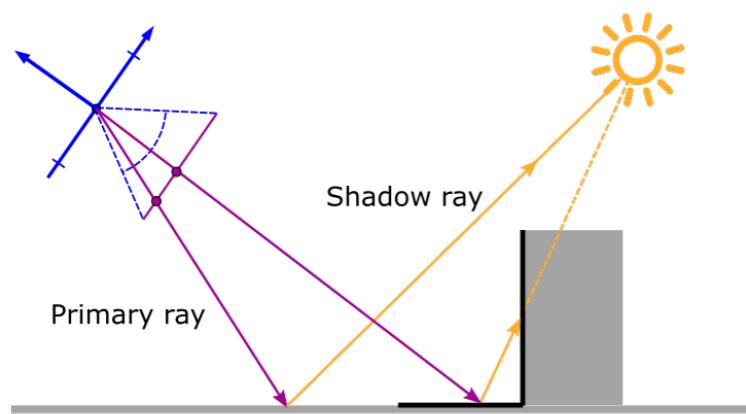
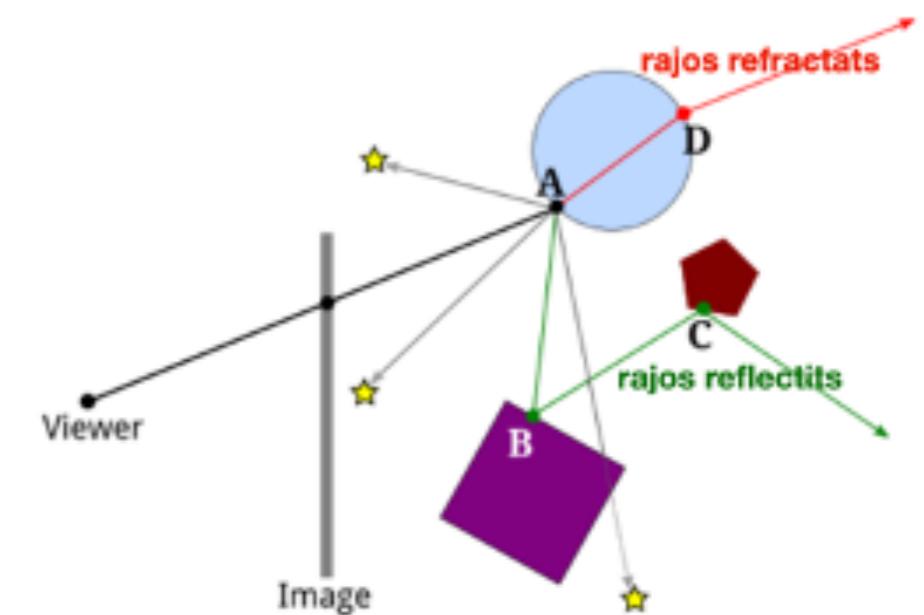


# Costos

- Cost de l'algorisme?

Suposem:

- Frame Buffer  $\text{numPixelsX} * \text{numPixelsY}$
- per cada pixel: **numRajosPrimaris**
- per cada raig primari: **numRajos Secundaris**
- num Llums
- numObjectes amb **cost intersecció + càcul ll.luminació**



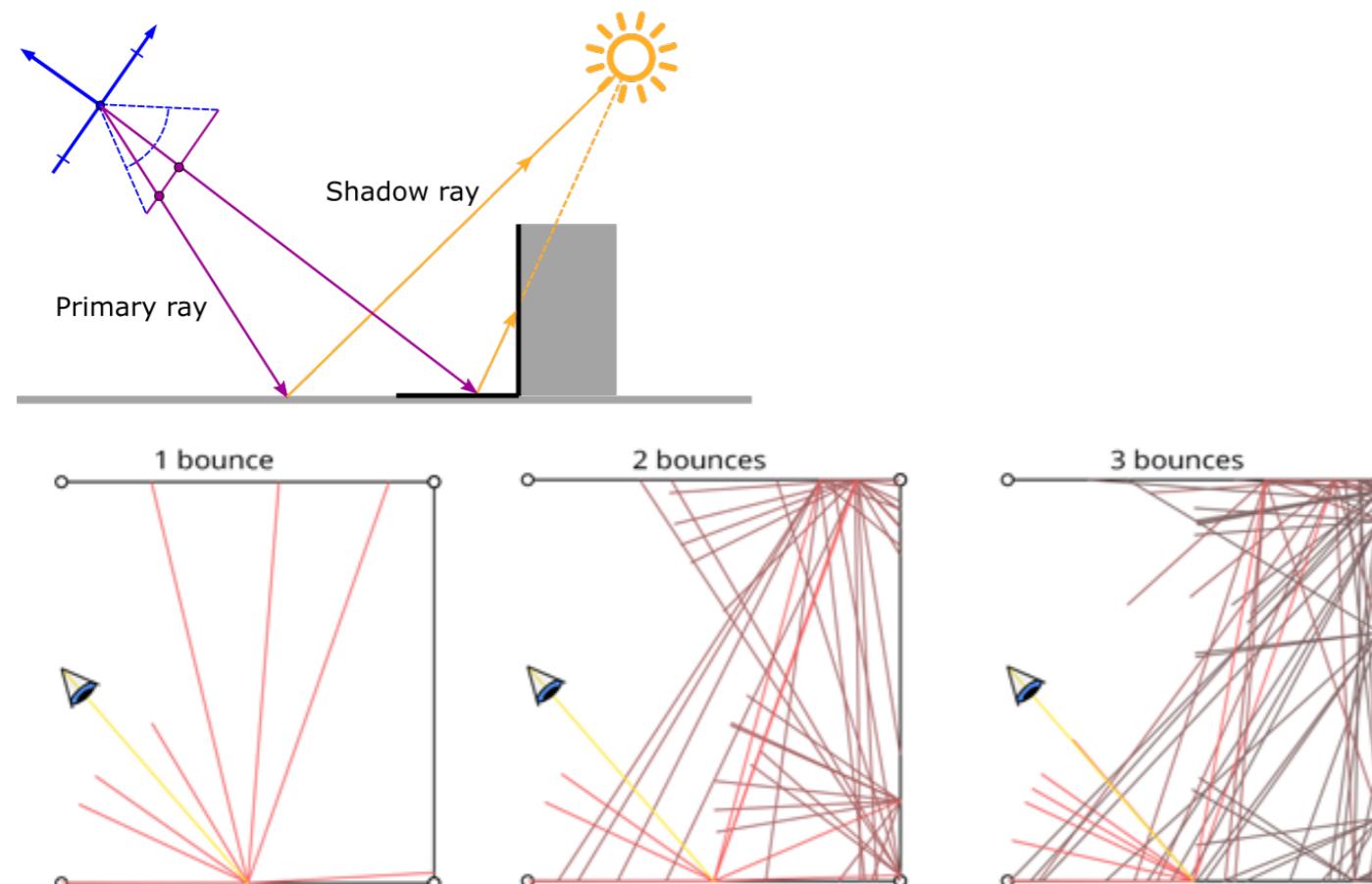
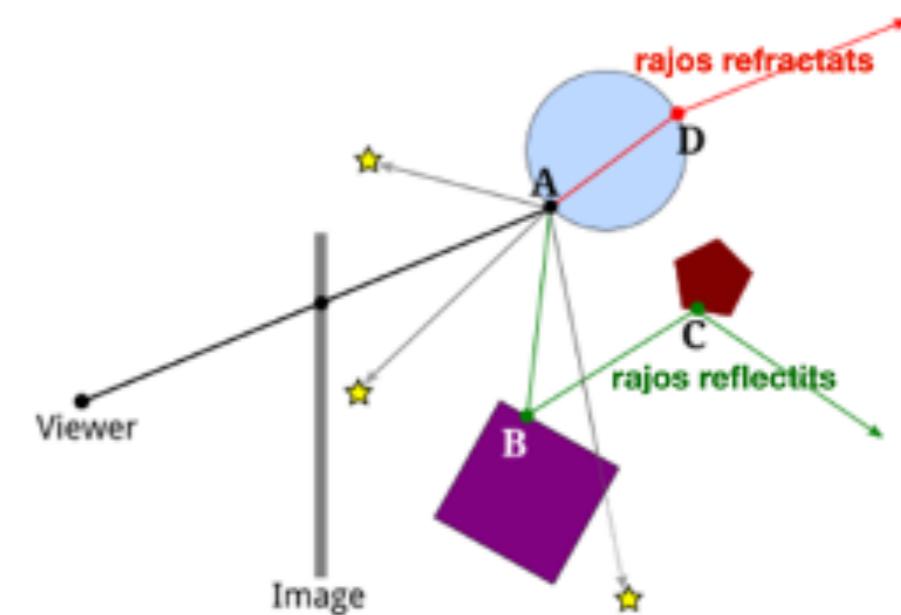
# Costos

- Cost de l'algorisme: Aprox.

`numPixelsX * numPixelsY *`

`( numRajosPrimaris * numRajos Secundaris * numLlums*`

**((numObjectes \* (cost intersecció + càcul ll.luminació))))**



Per a calcular interseccions:

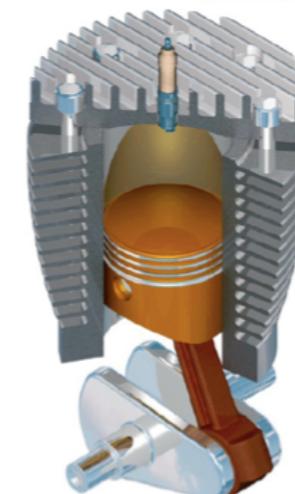
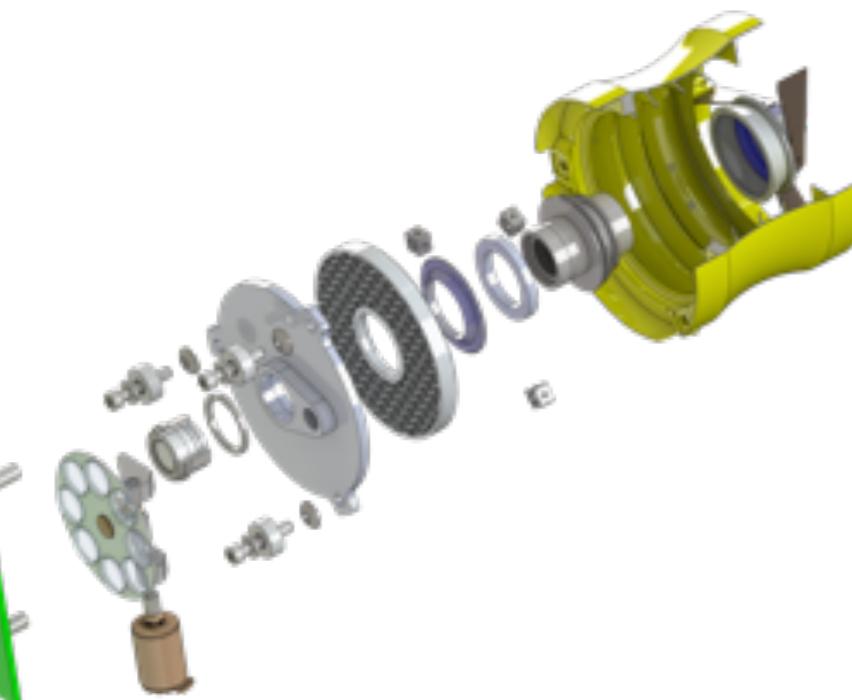
- En escenes simples: 75% temps total
- En escenes complexes: pot arribar al 95%

# 4. Interseccions

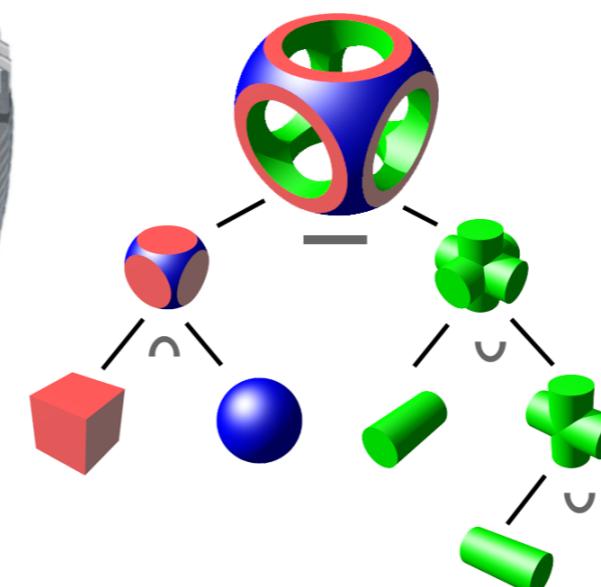
Una **escena** està formada per un conjunt d'objectes

- Com s'estructura?

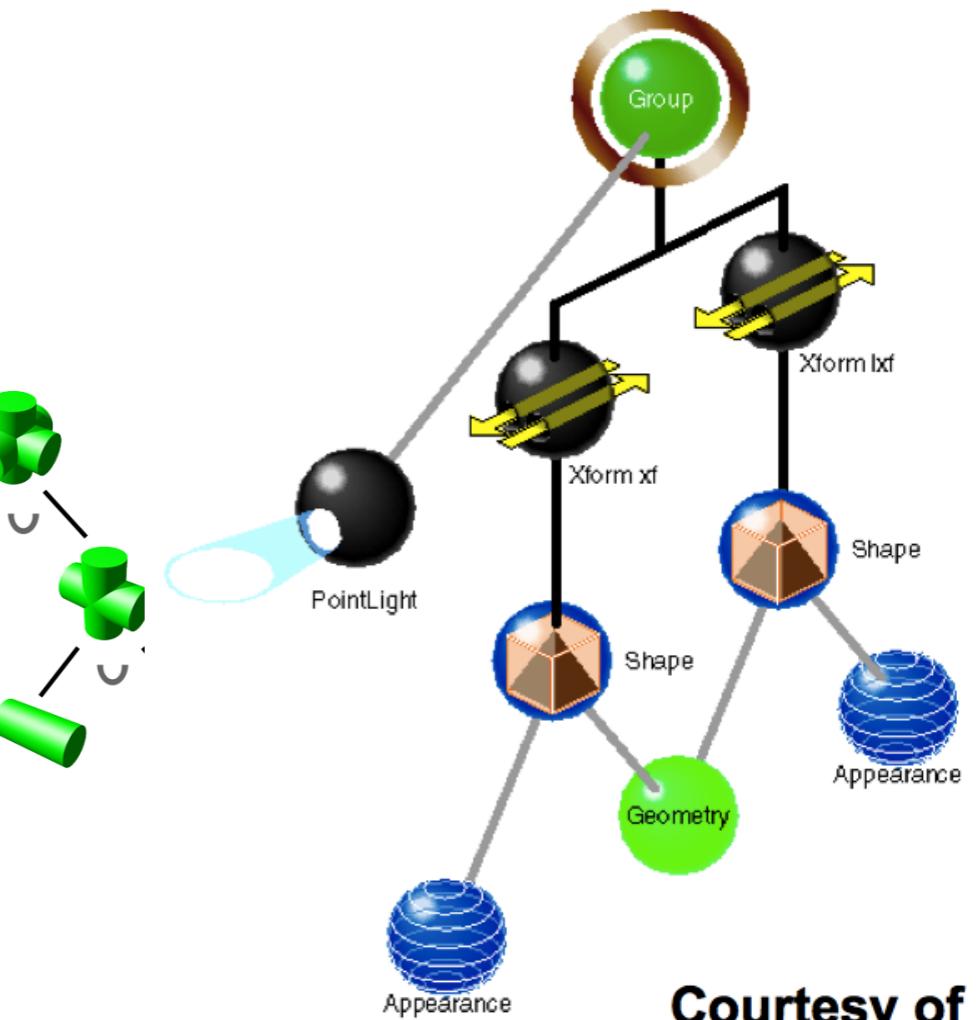
## Llista d'objectes



## Arbres CSG



## Scene Graph

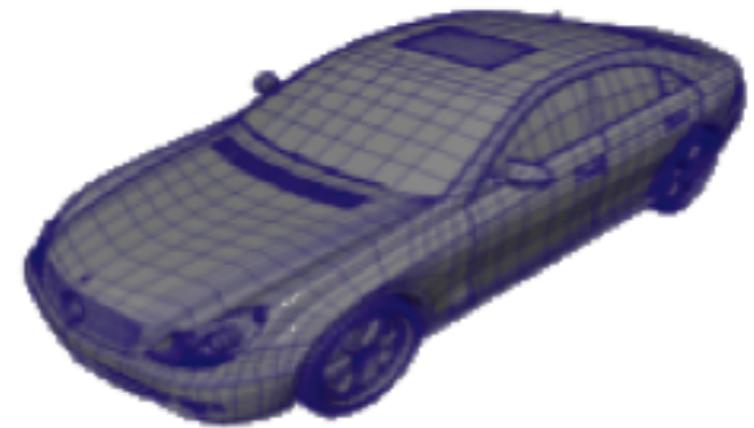
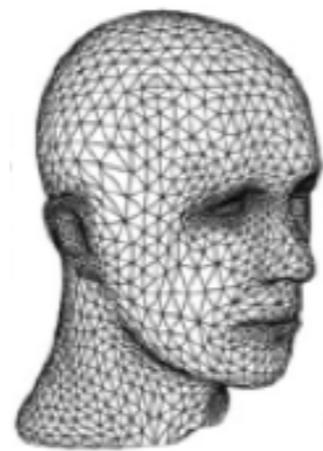


- Intersecció amb el raig?
  - Recorregut per tots els objectes i es troba la intersecció més propera amb  $t > 0$ , si l'escena només està formada per objectes superficials

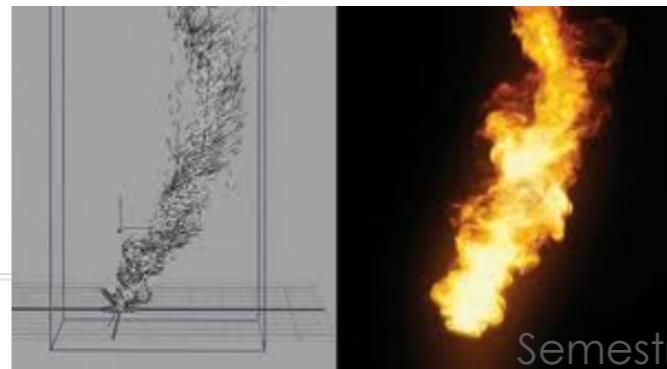
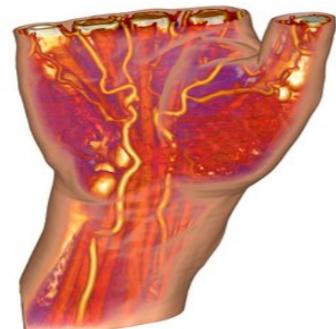
# 4. Interseccions

Tipus d'objectes:

- **Objectes superficials (models 2D)**: només representen la frontera de l'objecte amb l'exterior (Boundary -objects o BR-Objects)

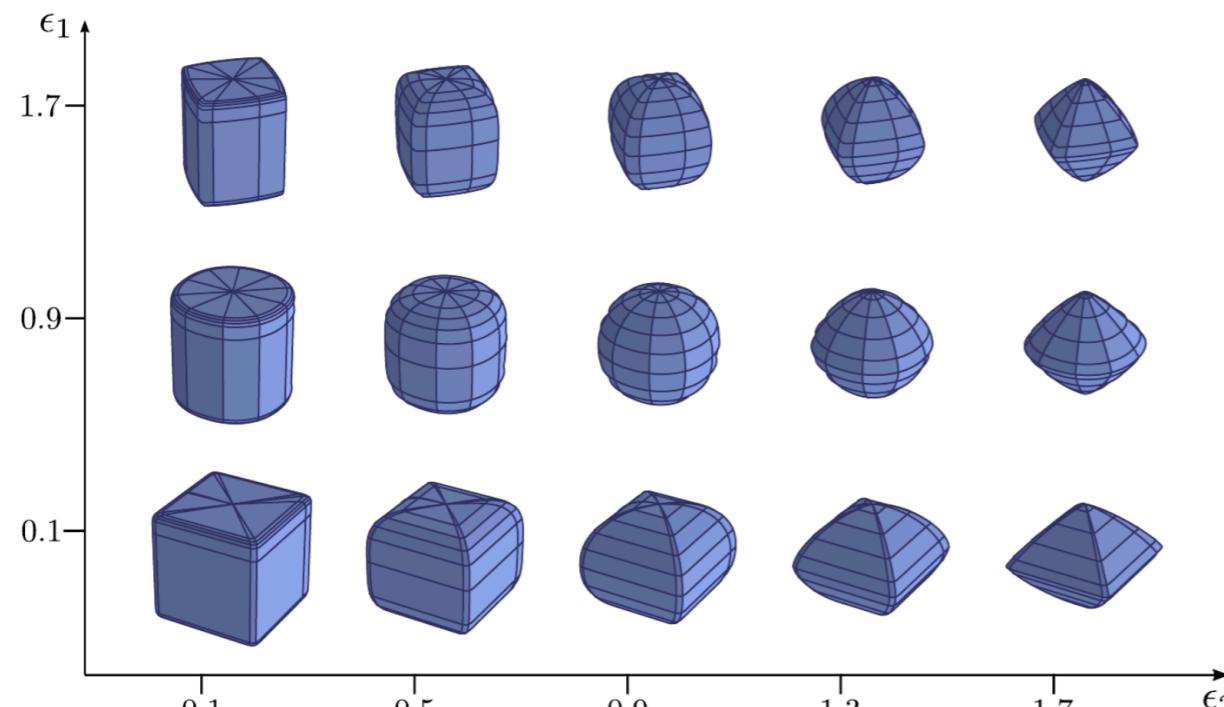
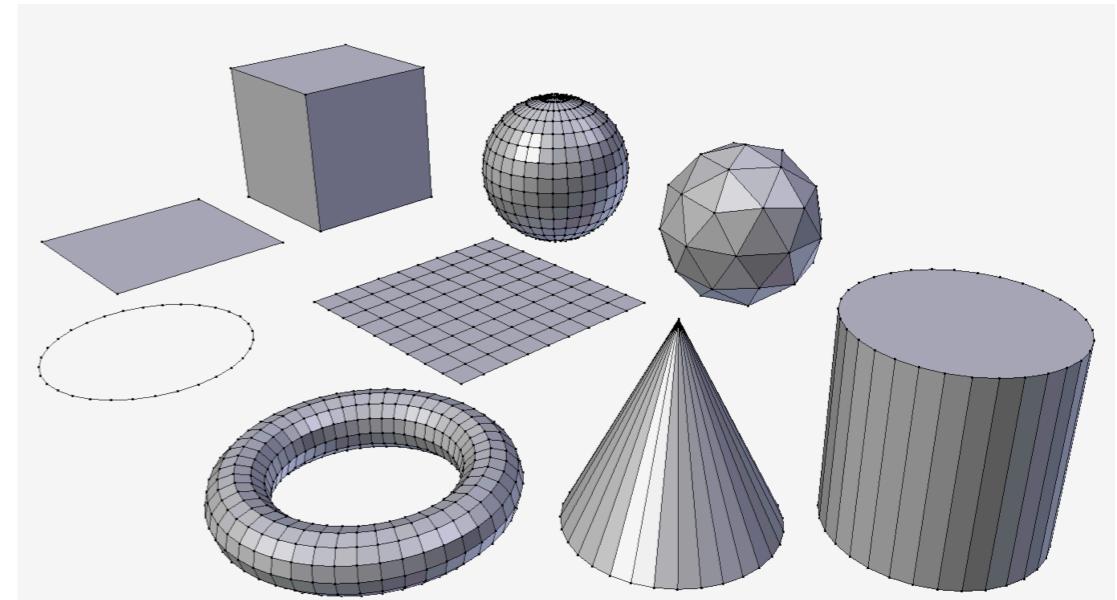


- **Objectes volumètrics (models 3D)**: es representa el volum intern dels objectes o característiques de l'espai com temperatura, densitat, etc.)

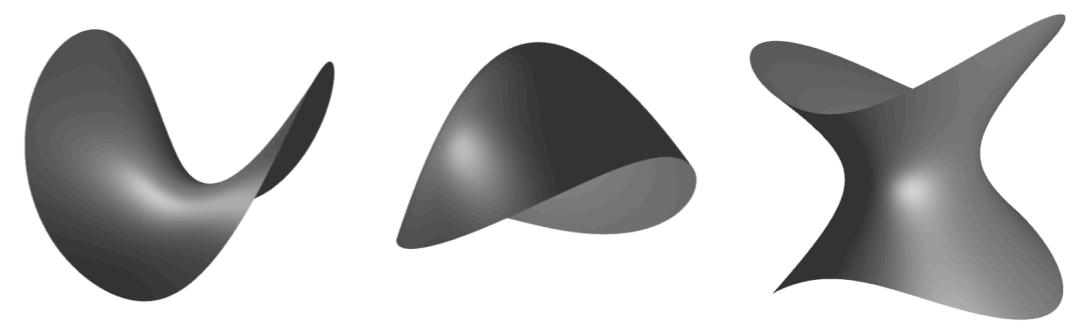


# 4. Interseccions

- Objectes paramètrics:
  - esferes
  - cilindres
  - torus
  - superquàdriques



<https://demonstrations.wolfram.com/Superquadrics/>



superfícies implícites de segon ordre

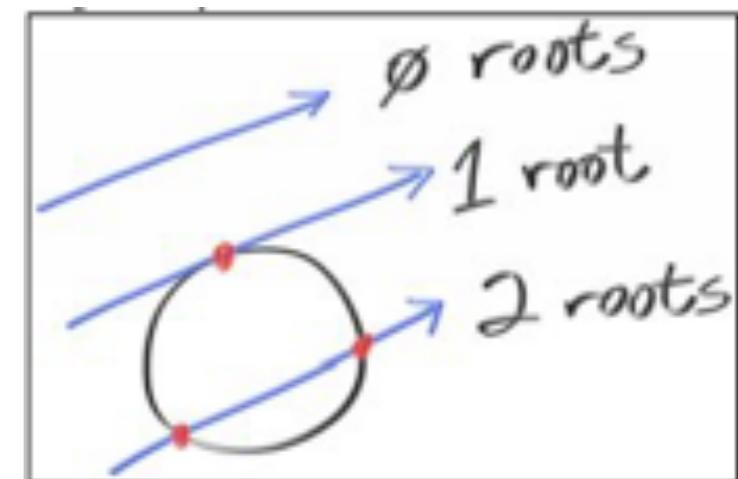
<http://www.dgp.toronto.edu/~lessig/icaster/icaster.html>

$$\left( \left( \frac{x}{\alpha_1} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{y}{\alpha_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{\alpha_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

# Objectes superficials: paramètrics

Esfera:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r^2$$



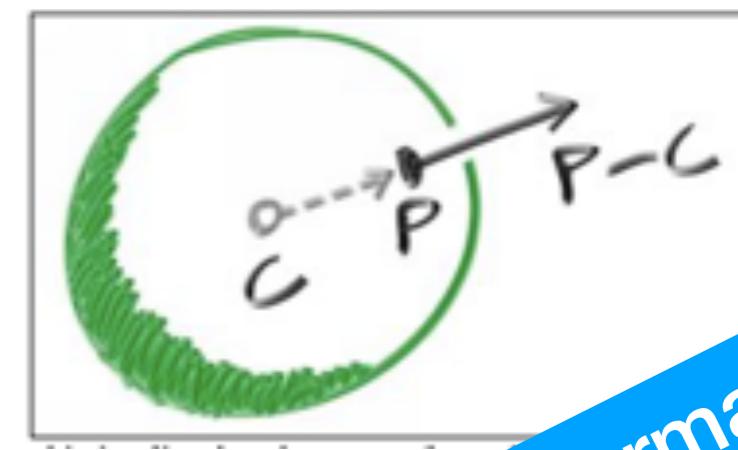
Recta:

$$p = p_o + t * \vec{v}$$

$$p_x = p_{x_0} + t * v_x$$

$$p_y = p_{y_0} + t * v_y$$

$$p_z = p_{z_0} + t * v_z$$

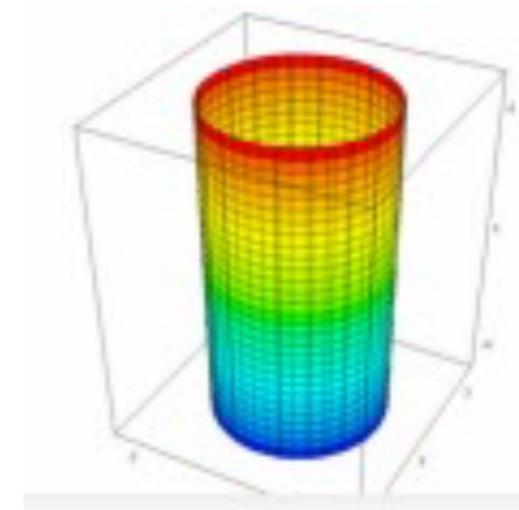


normal?

I un cilindre?

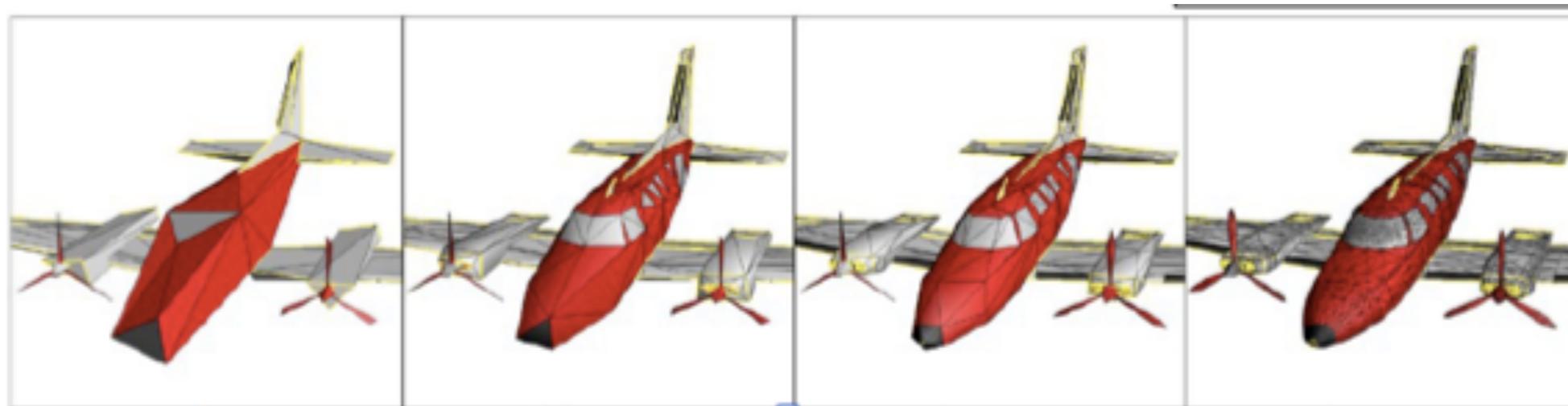
$$(x - x_c)^2 + (z - z_c)^2 = r^2$$

I les tapes?



# 4. Interseccions

- **Malles poligonals:**



15,000 triangles



15,000 triangles



1 million triangles

[1] Veure secció 2.4.4 del llibre [Angel2011]

# Triangle: intersecció?

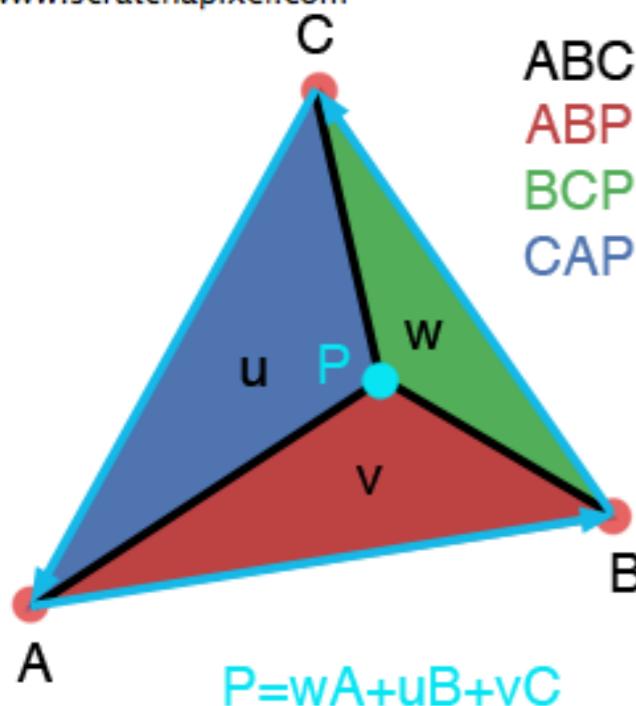
Com es calcula la intersecció del raig amb cada triangle?

1. Càlcul de la intersecció amb el pla del triangle:

- Pla amb normal  $n$
- Pla que passa per un vèrtex del triangle

2. Comprovació si el punt és interior al triangle: usant  
**coordenades baricèntriques** ( $u, v, w$ ) <https://www.geogebra.org/m/ZuvmpJmy>

© www.scratchapixel.com



$$u = \frac{\text{TriangleCAP Area}}{\text{TriangleABC Area}}$$

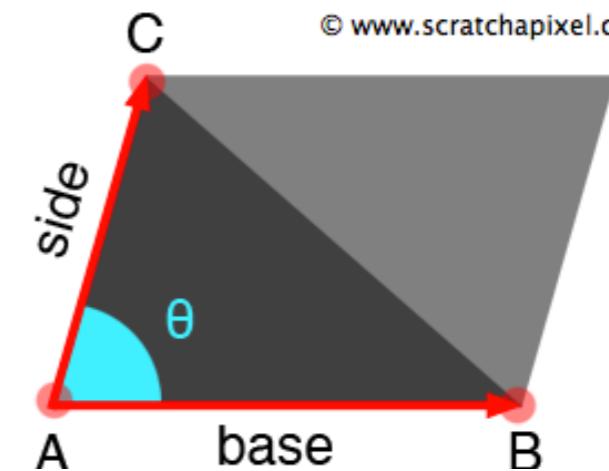
$$v = \frac{\text{TriangleABP Area}}{\text{TriangleABC Area}}$$

$$w = \frac{\text{TriangleBCP Area}}{\text{TriangleABC Area}}$$

$$\boxed{P = wA + uB + vC}$$

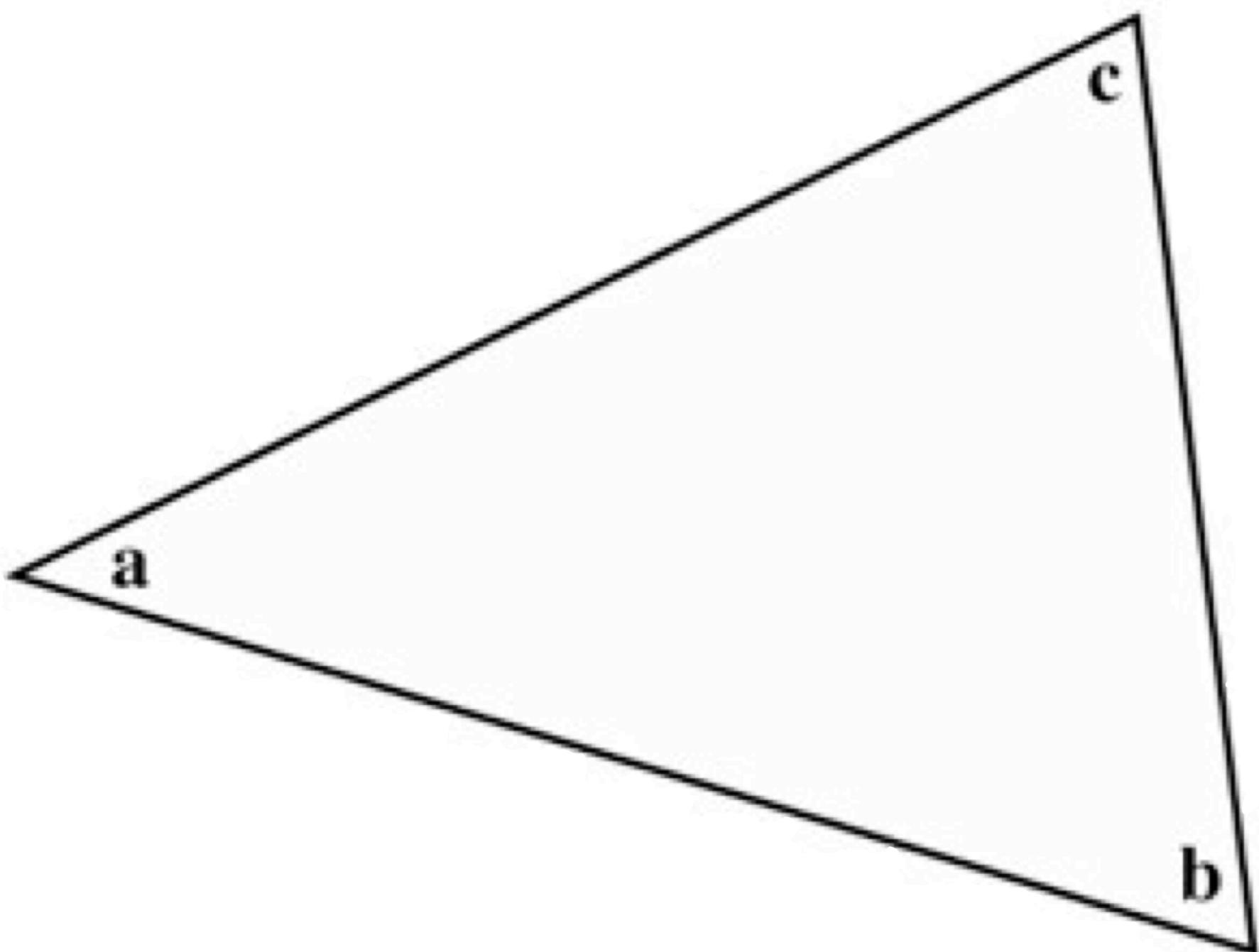
on  $u+v+w = 1$  i  $0 < u, v, w < 1$

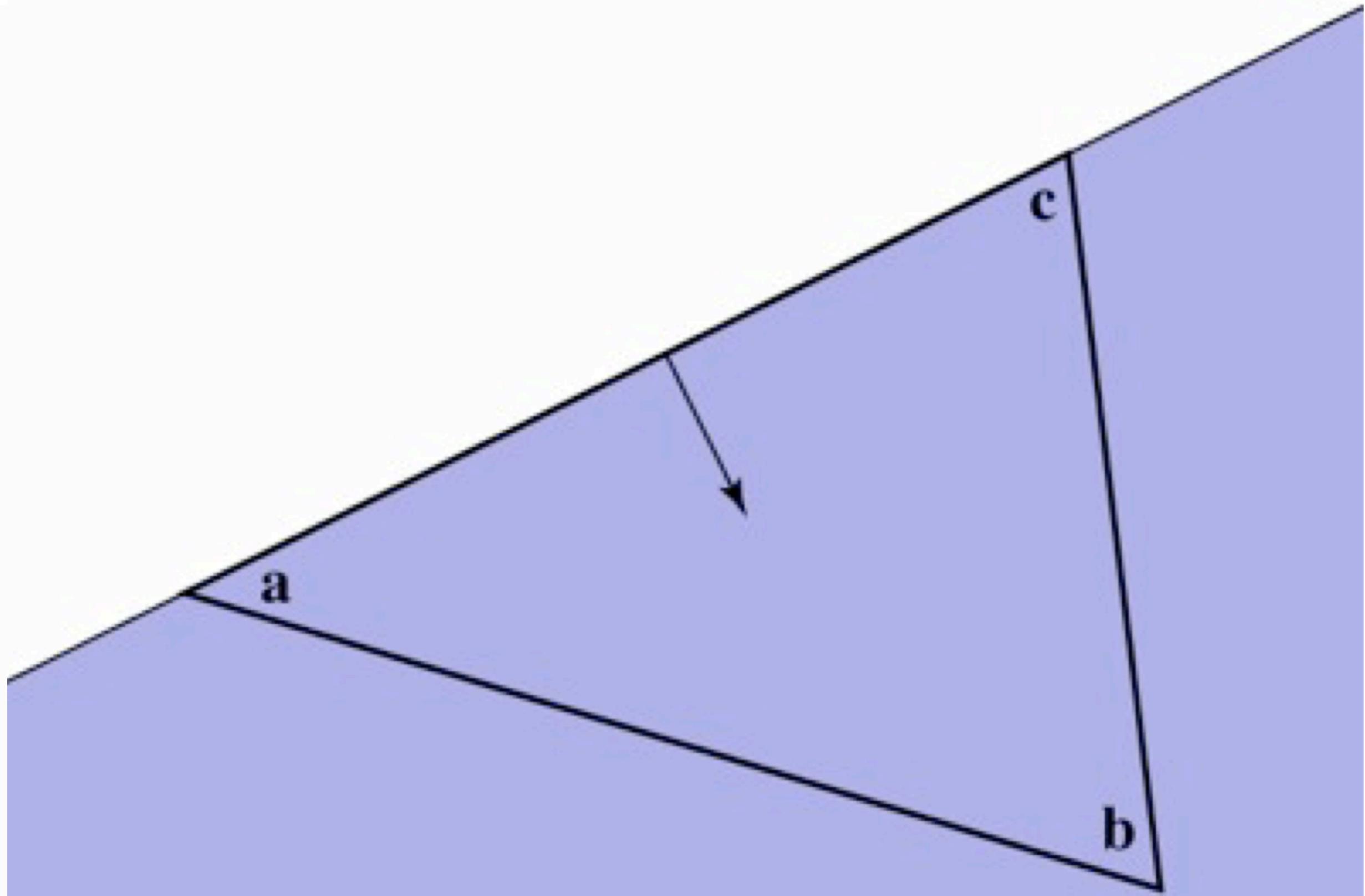
© www.scratchapixel.com

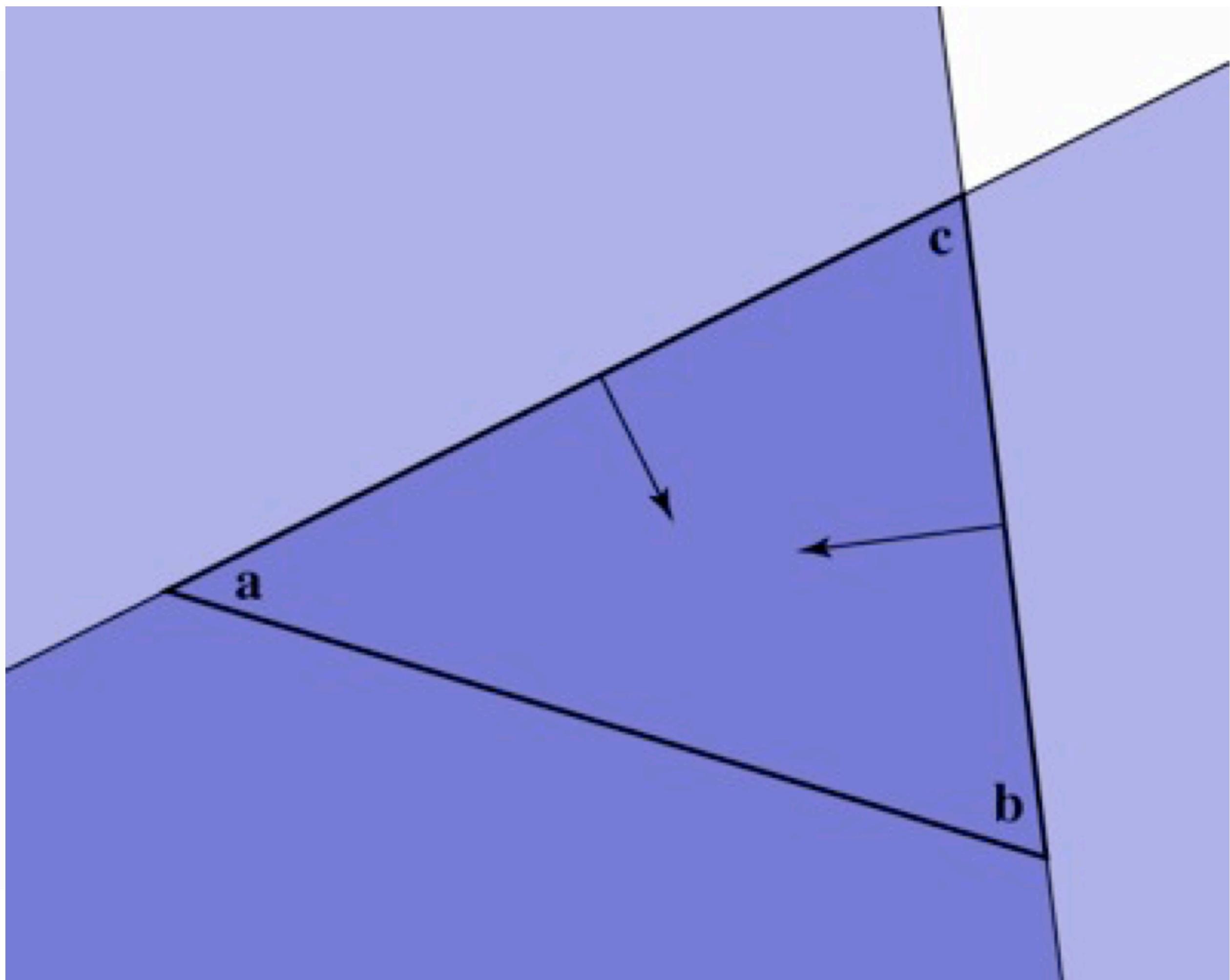


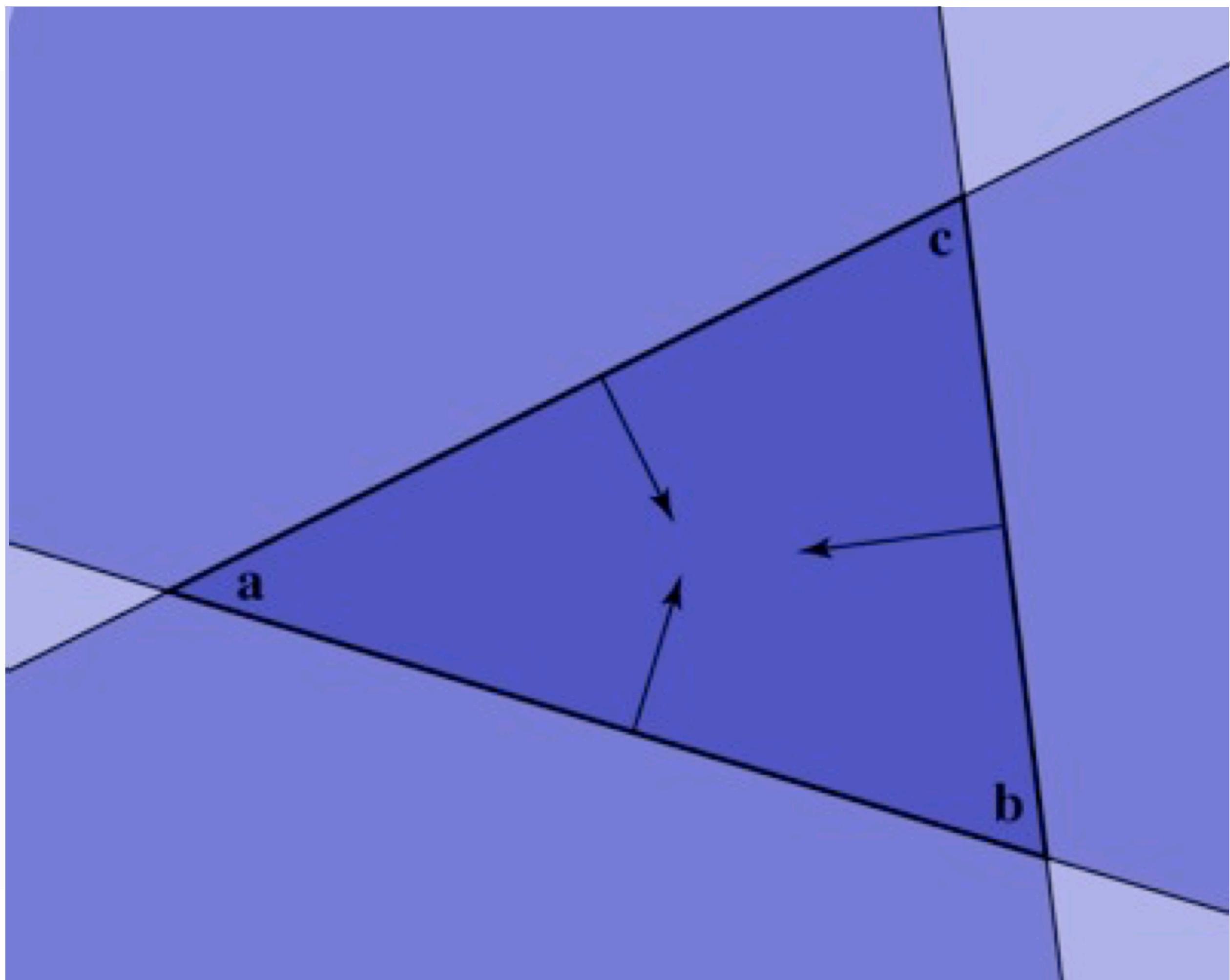
$$\text{Triangle}_{\text{area}} = \frac{\|(B - A)\| * \|(C - A)\| \sin(\theta)}{2}$$

# Triangle: intersecció?

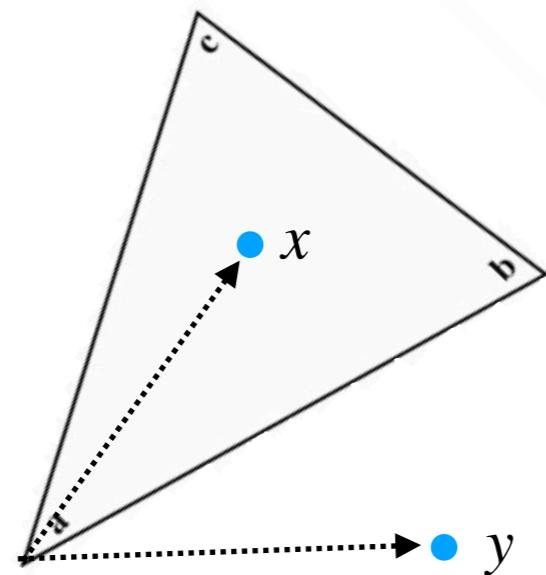








# Triangle: intersecció?



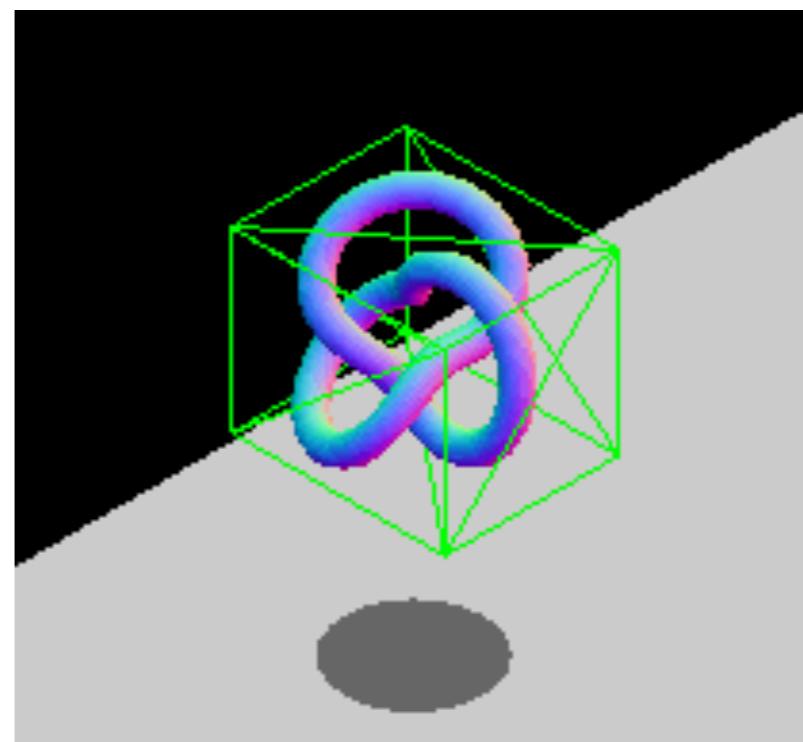
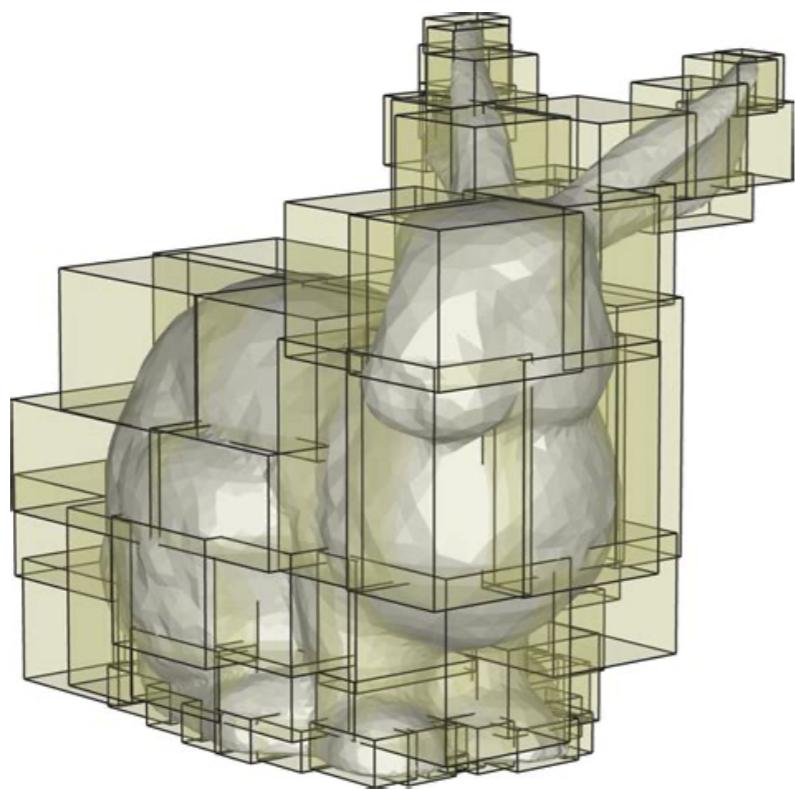
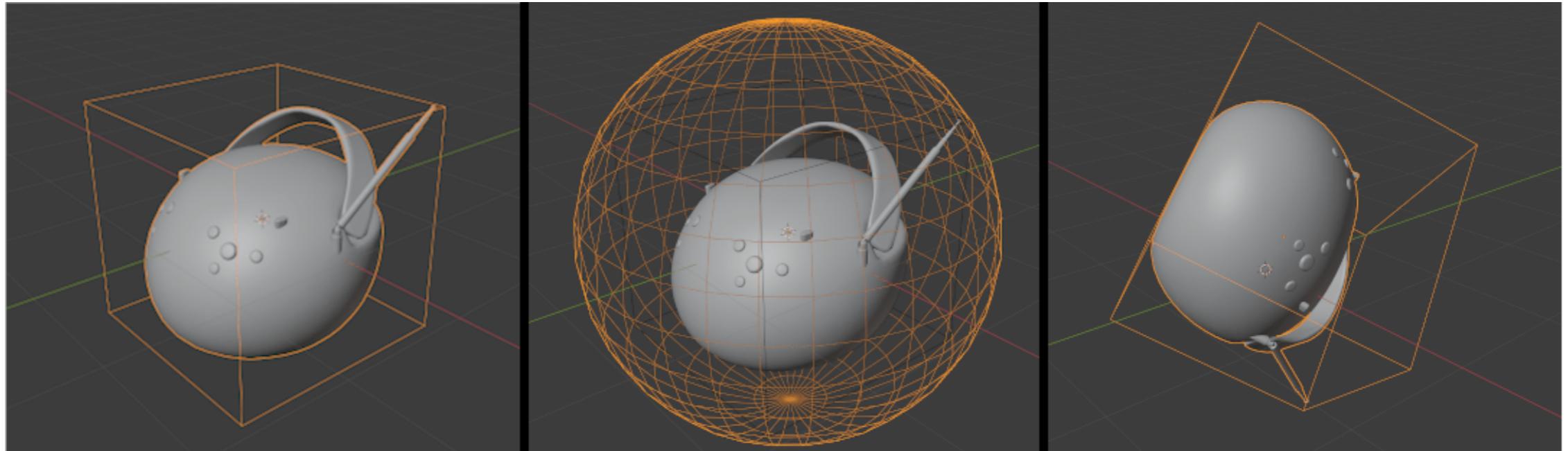
$$(\mathbf{b} - \mathbf{a}) \times (\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} > 0$$

$$(\mathbf{c} - \mathbf{b}) \times (\mathbf{x} - \mathbf{b}) \cdot \mathbf{n} > 0$$

$$(\mathbf{a} - \mathbf{c}) \times (\mathbf{x} - \mathbf{c}) \cdot \mathbf{n} > 0$$

<https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/ray-triangle-intersection-geometric-solution.html>

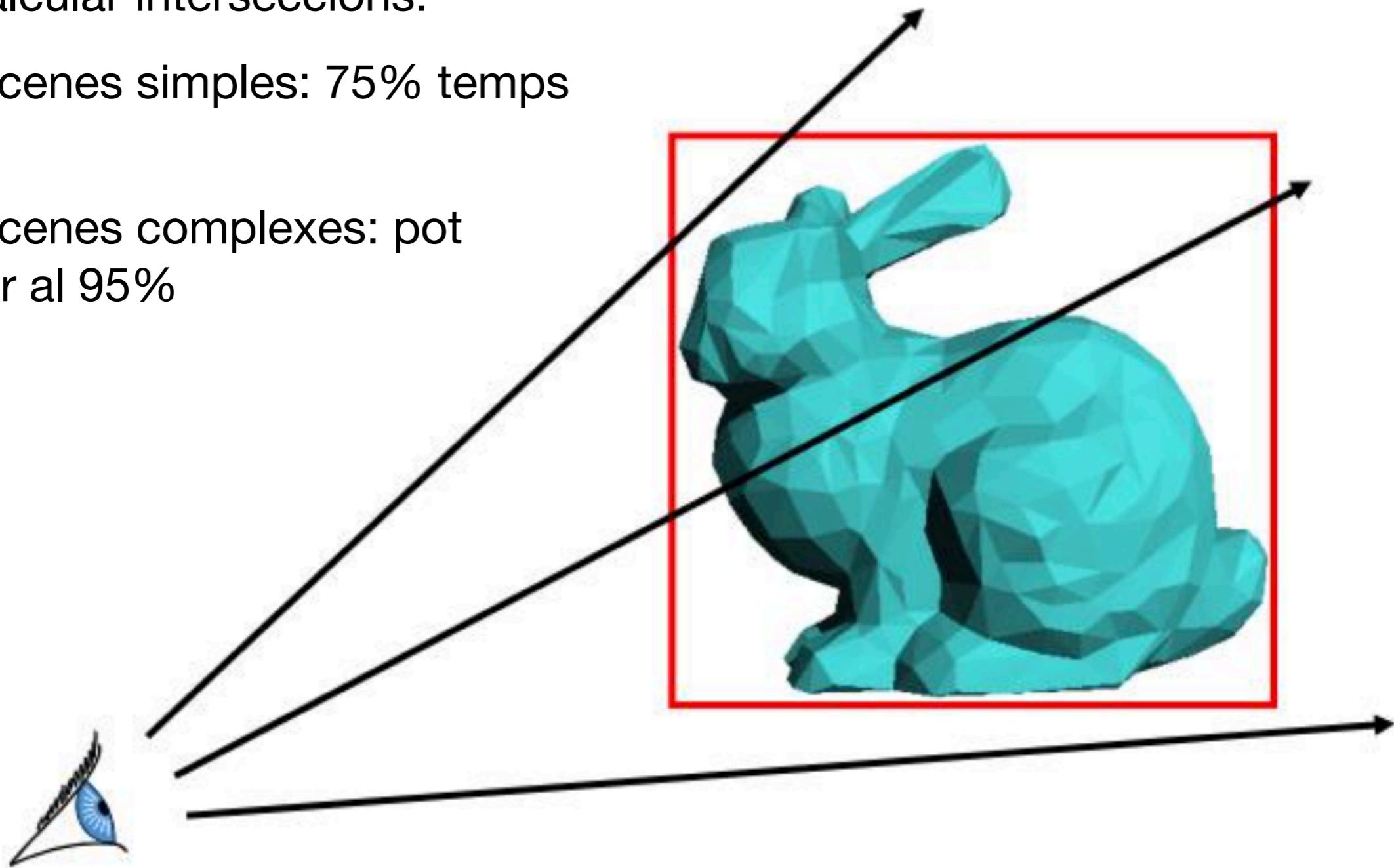
# 4. Interseccions



# 4. Interseccions

Per a calcular interseccions:

- En escenes simples: 75% temps total
- En escenes complexes: pot arribar al 95%



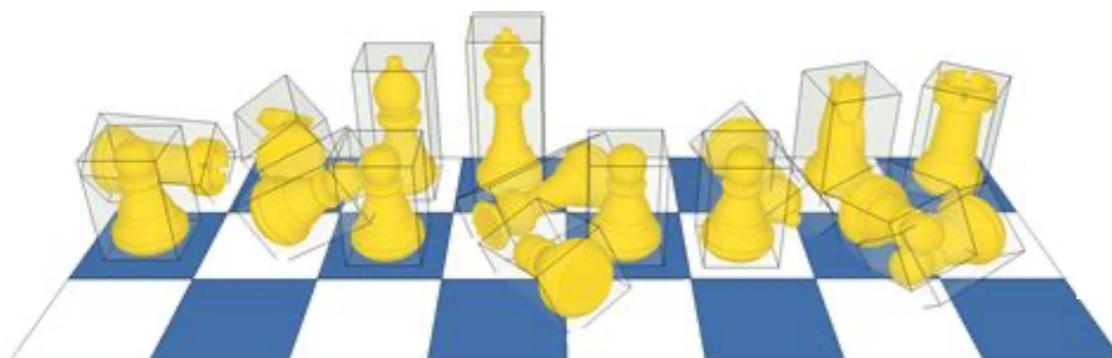
# Acceleració de les interseccions raig-objecte



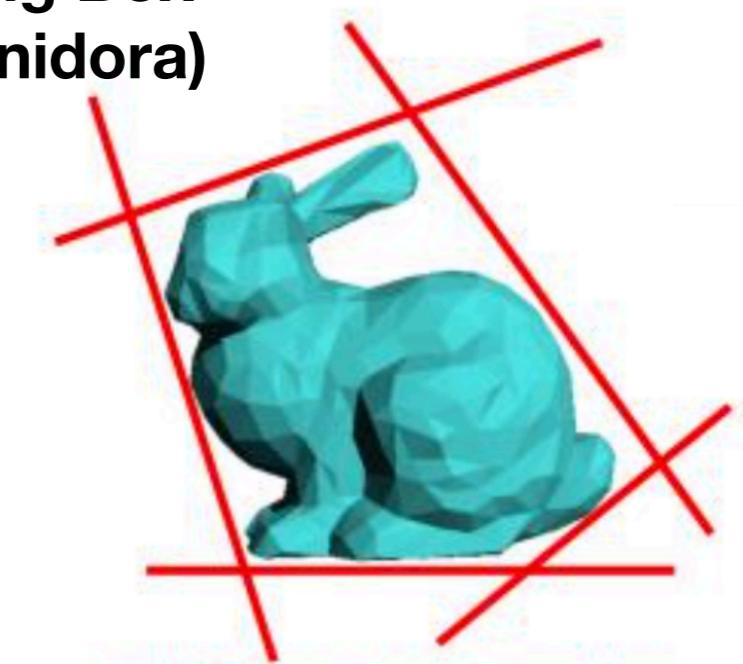
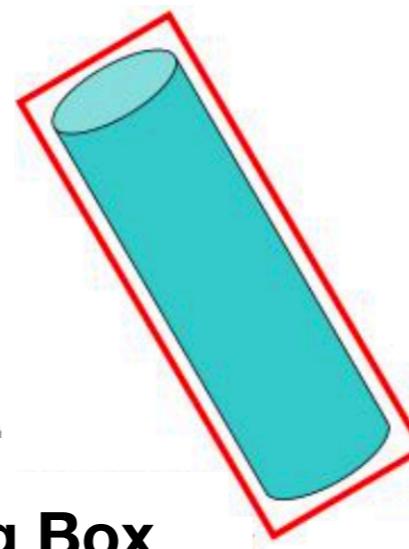
**Bounding sphere**  
**(Esfera mínima contenidora)**



**Axis-Aligned Bounding Box**  
**(Capsa mínima contenidora)**



**Object-Oriented Bounding Box**



**Bounding Half-Spaces**  
**(plans arbitraris)**

# Capsa mínima contenidora (AABB)

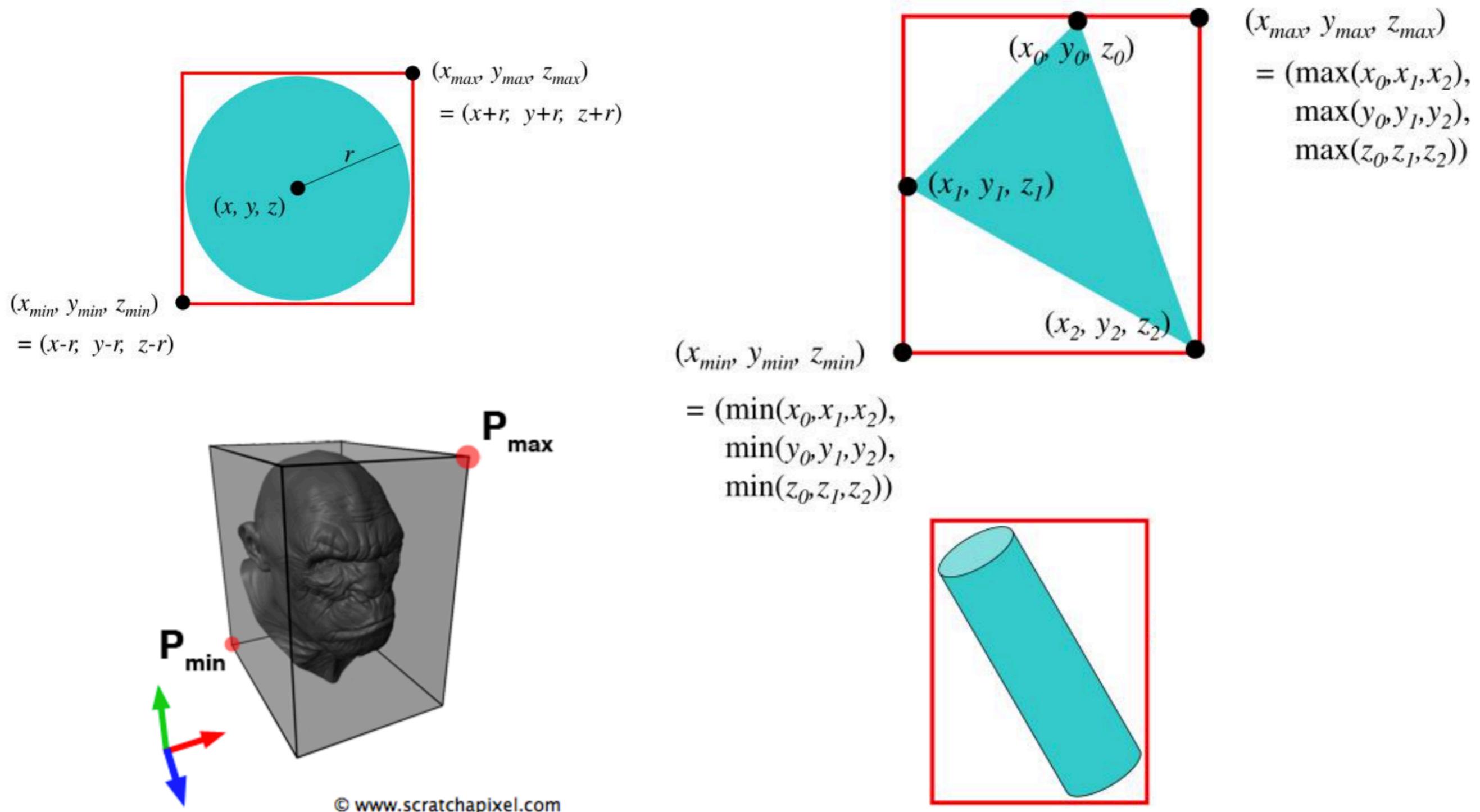
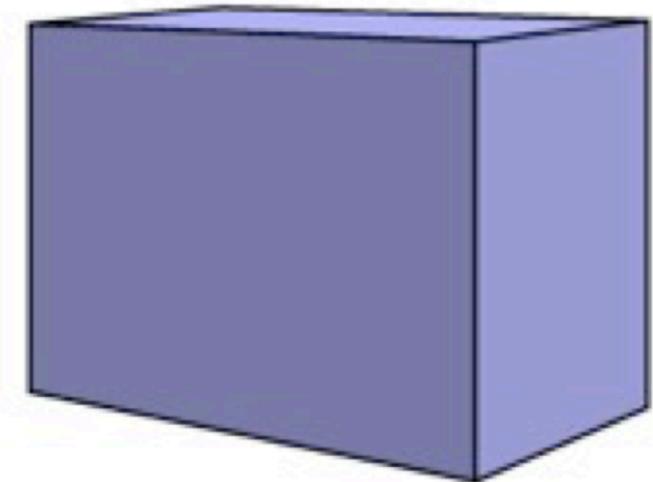
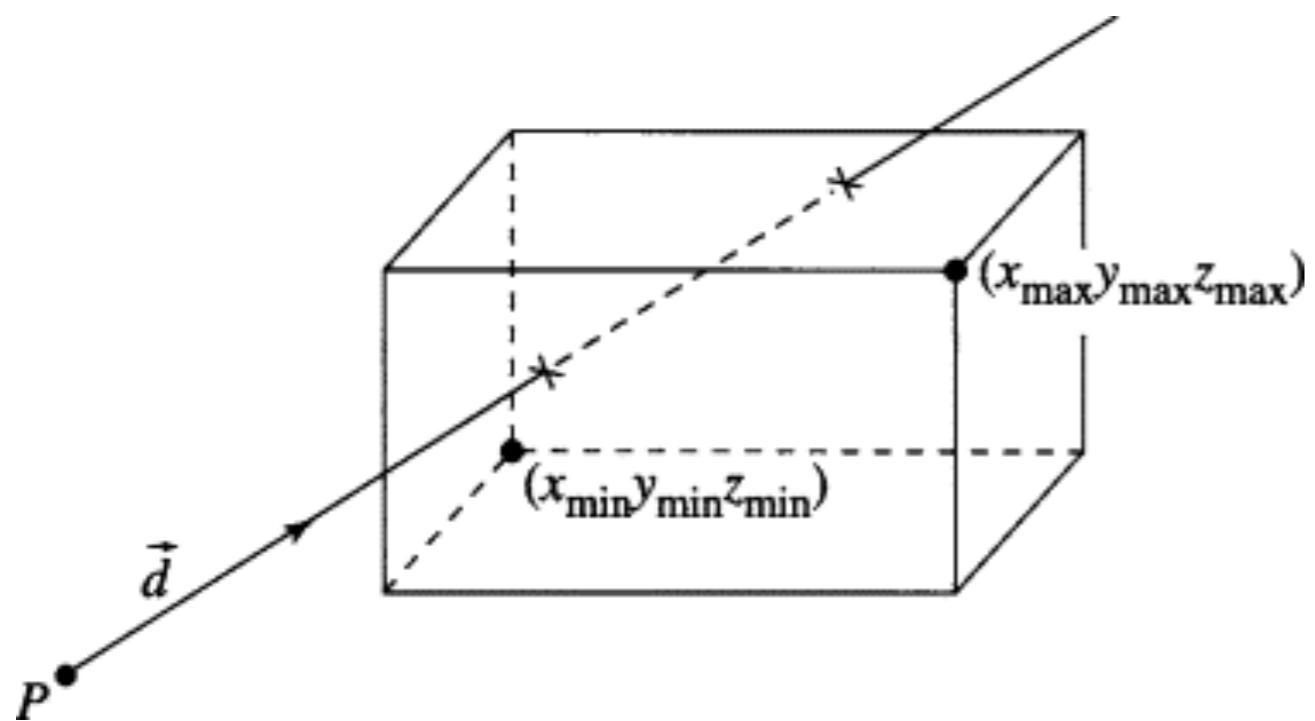


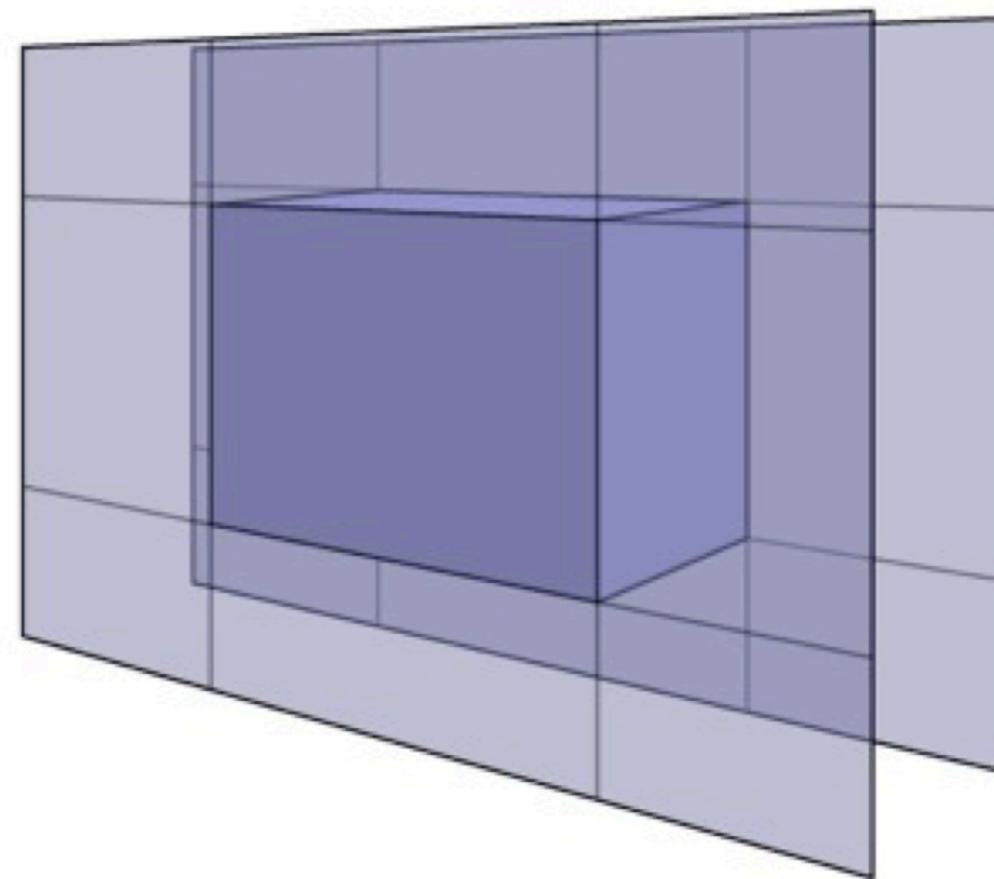
Figure 3: the bounding box of the Ape model  
(courtesy of [Kenichi Nishida](#)).

# AA-Box: intersecció?

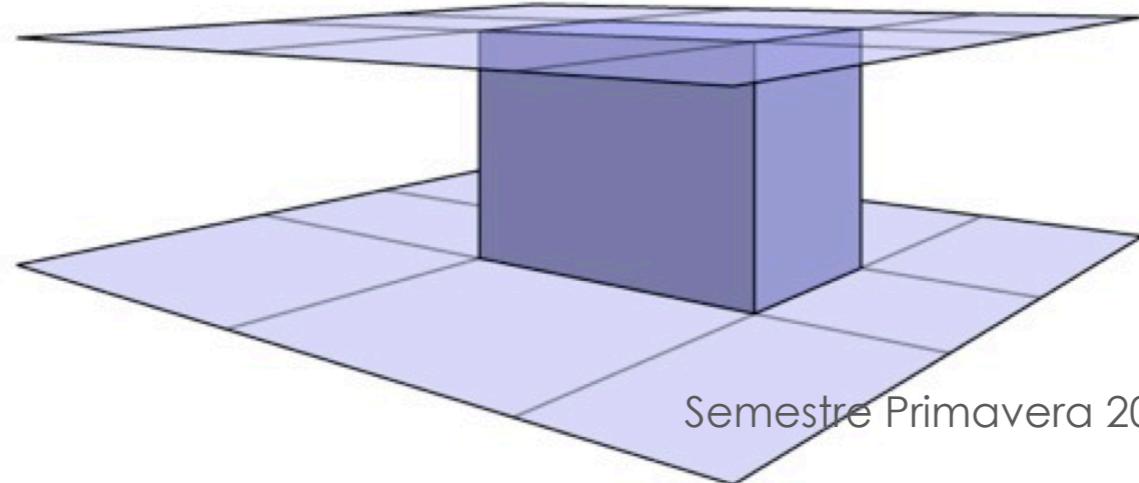
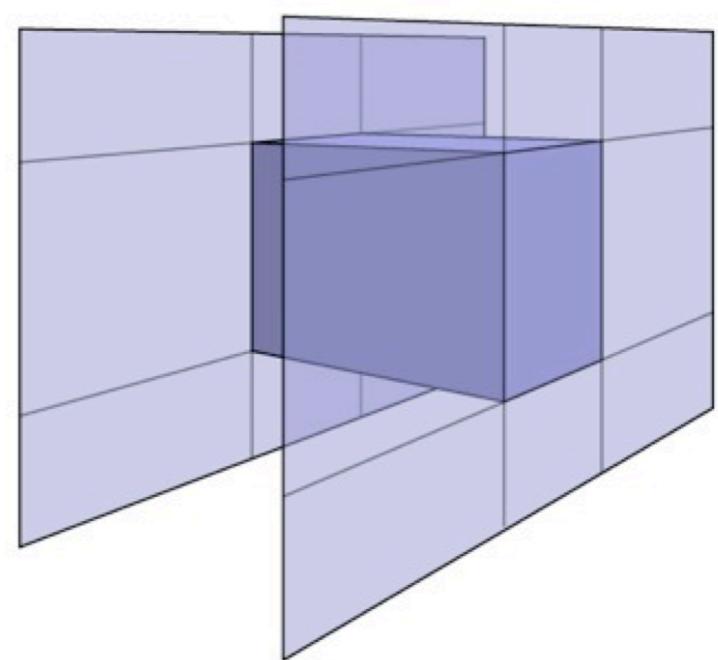
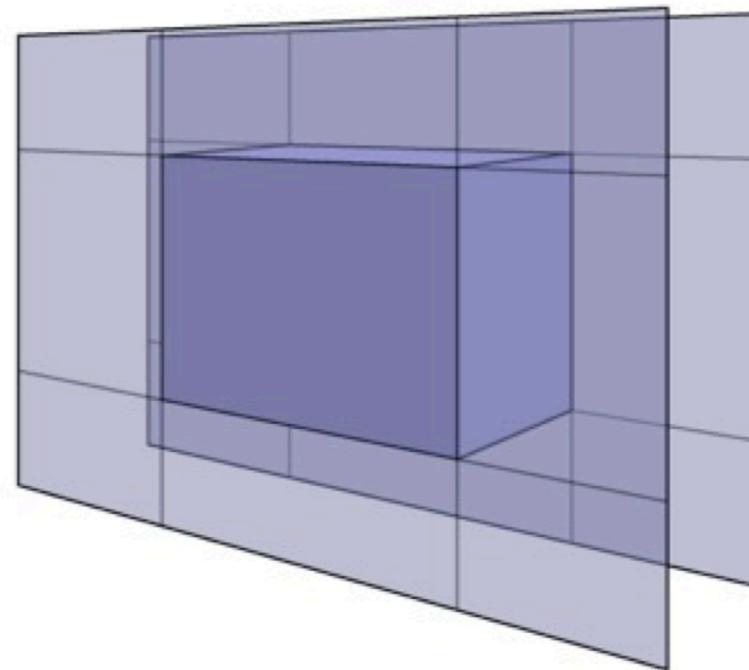
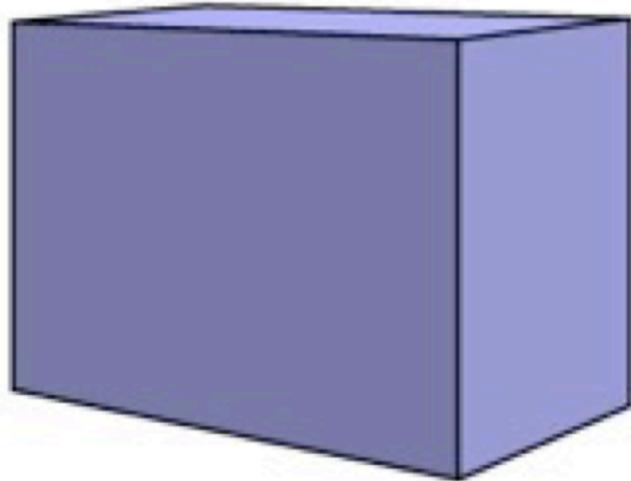
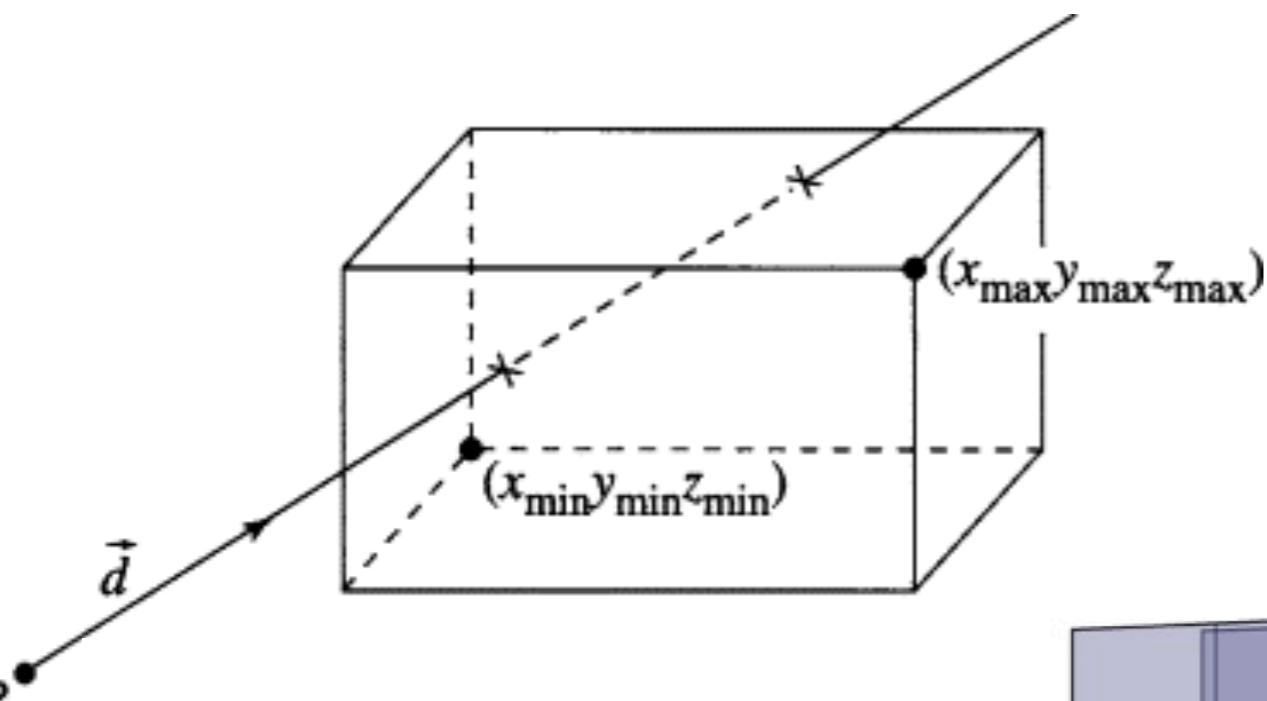


Possibilitats:

1. Intersecar amb cada cara individualment
2. Intersecar slabs (una box és la intersecció de 6 slabs)



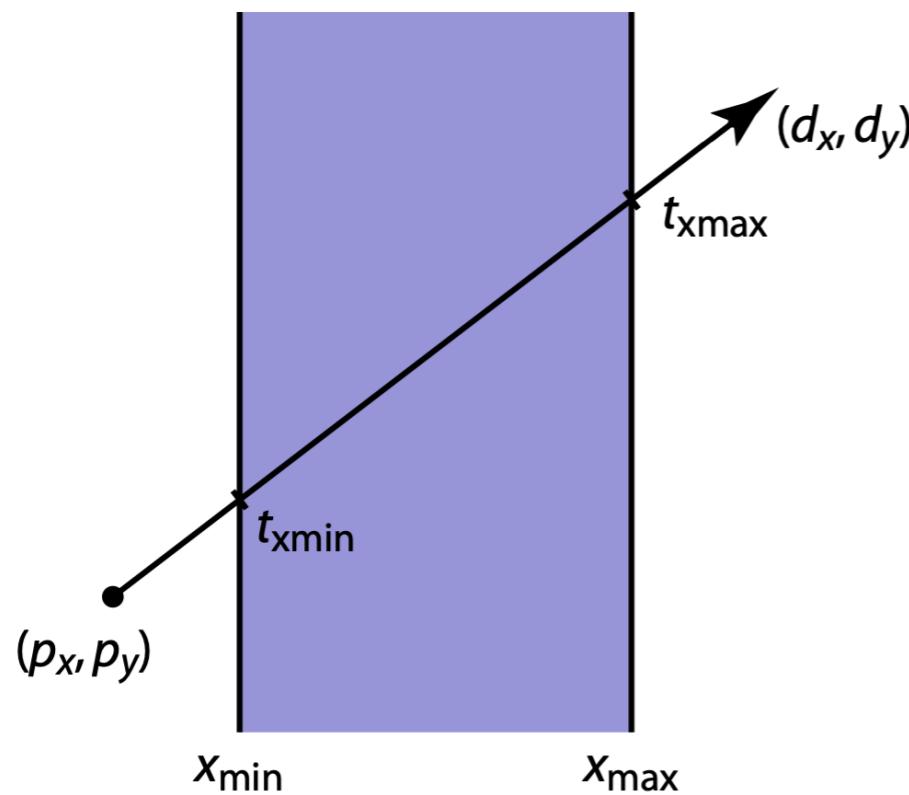
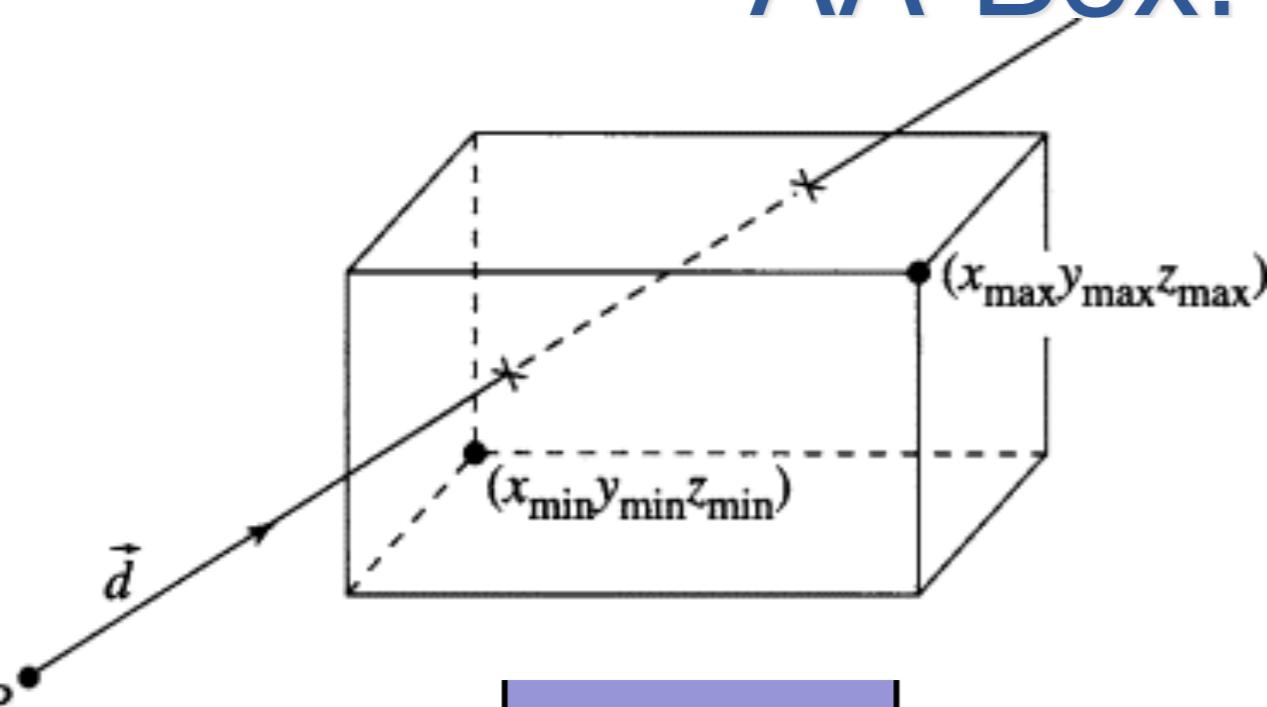
# AA-Box: intersecció?



Possibilitats:

1. Intersecar amb cada cara individualment
2. Intersecar slabs (una box és la intersecció de 6 slabs)

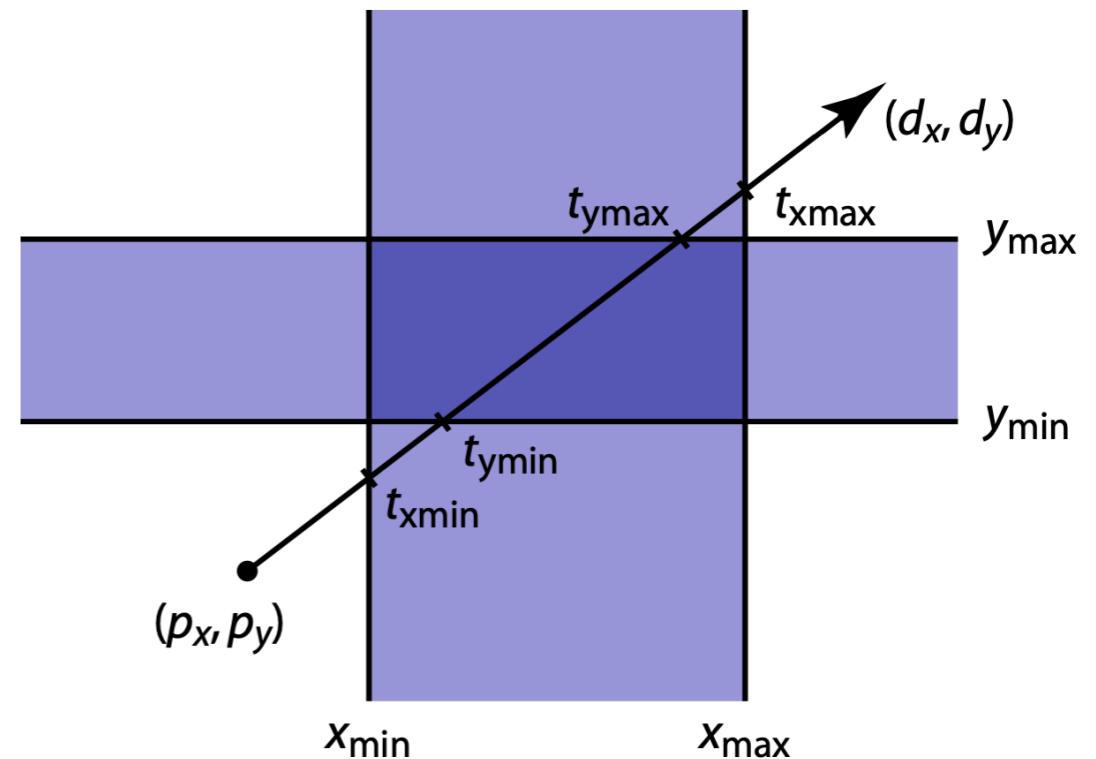
# AA-Box: intersecció?



$$\begin{aligned} x_{\min} &= p_x + t_{\min} \vec{d} & t_{\text{enter}_x} &= \min(t_{\min_x}, t_{\max_x}) \\ t_{\min_y} &= (x_{\min} - p_x) / \vec{d} & t_{\text{exit}_x} &= \max(t_{\min_x}, t_{\max_x}) \\ t_{\max_y} &= (x_{\max} - p_x) / \vec{d} \end{aligned}$$

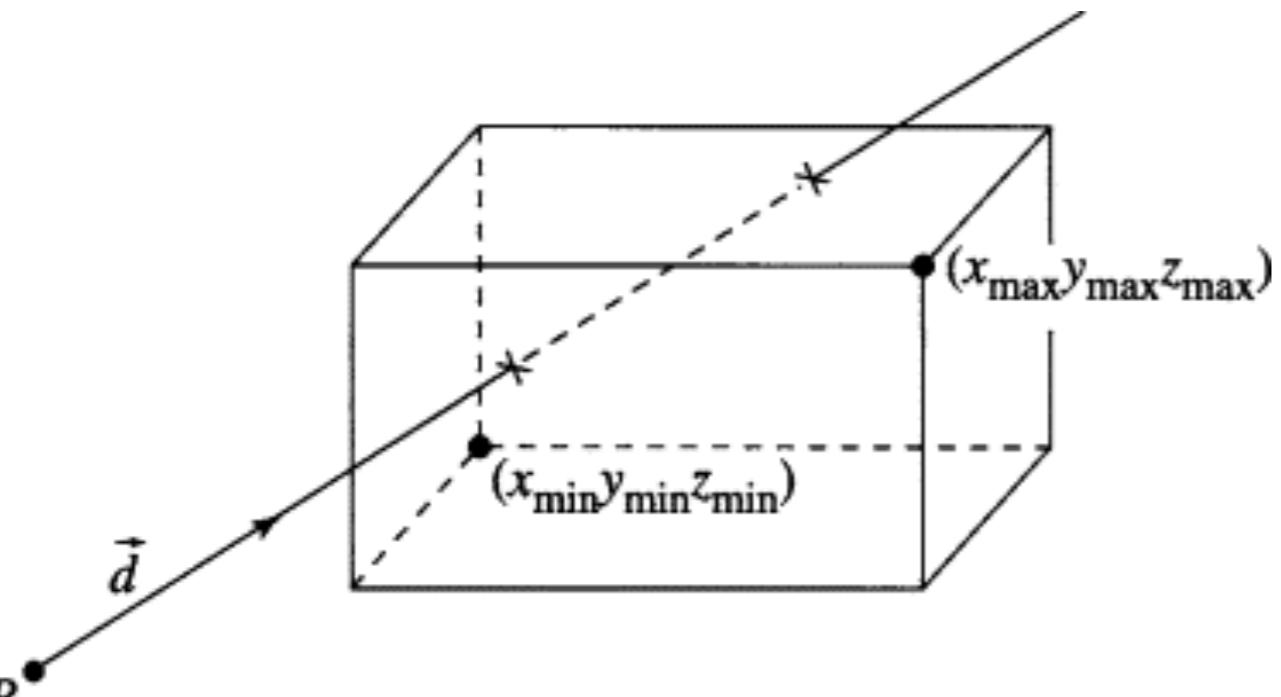
Possibilitats:

1. Intersecar amb cada cara individualment
2. **Intersecar slabs (una box és la intersecció de 6 slabs)**



$$\begin{aligned} t_{\text{enter}_y} &= \min(t_{\min_y}, t_{\max_y}) \\ t_{\text{exit}_y} &= \max(t_{\min_y}, t_{\max_y}) \end{aligned}$$

# AA-Box: intersecció?



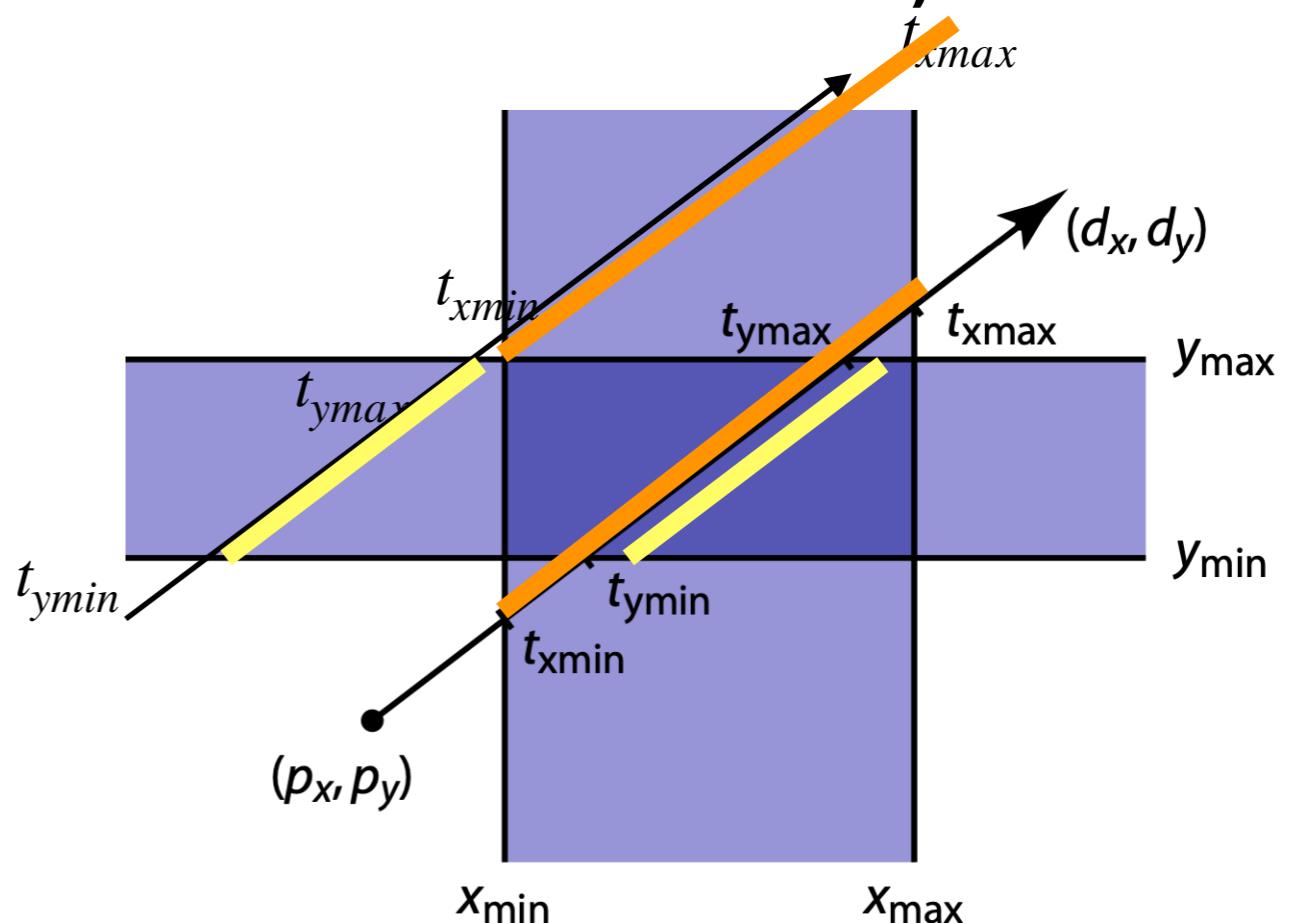
$$t_{\text{enter}} = \max(t_{\text{enter}_x}, t_{\text{enter}_y})$$

$$t_{\text{exit}} = \min(t_{\text{exit}_x}, t_{\text{exit}_y})$$

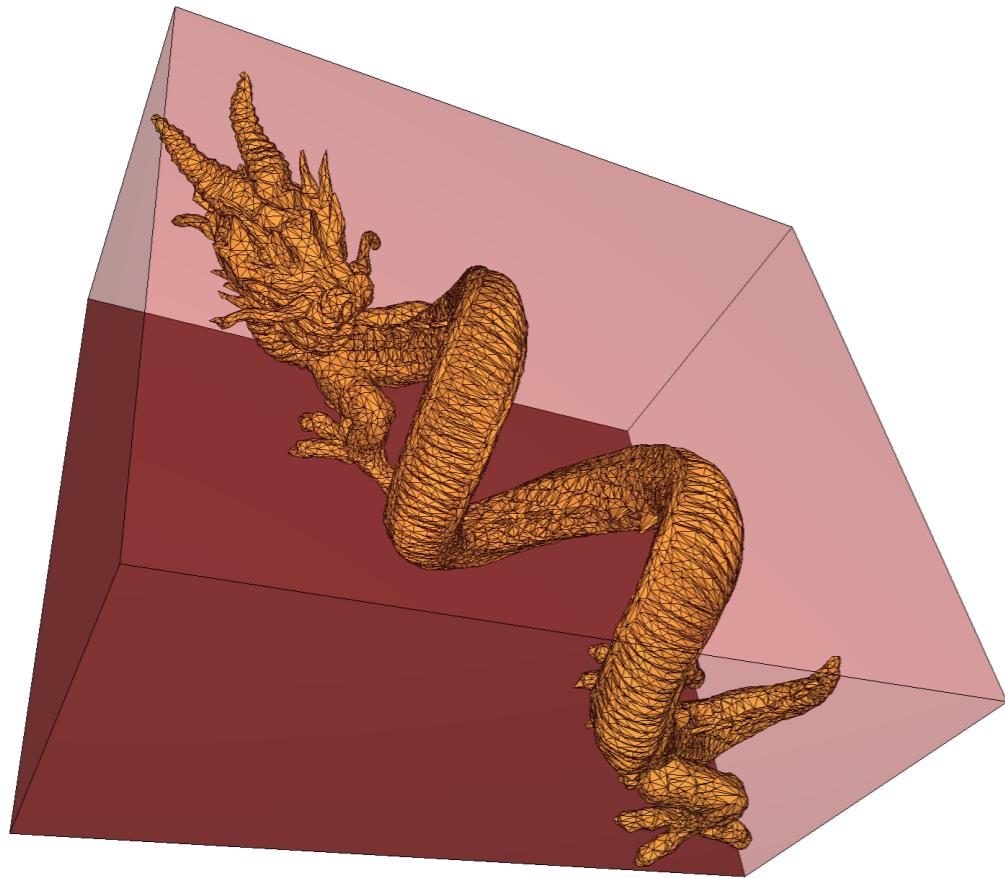
hit iff  $t_{\text{enter}} < t_{\text{exit}}$

Possibilitats:

1. Intersecar amb cada cara individualment
2. **Intersecar slabs (una box és la intersecció de 6 slabs)**



# AA-Box vs Object-Aligned Boundig Box

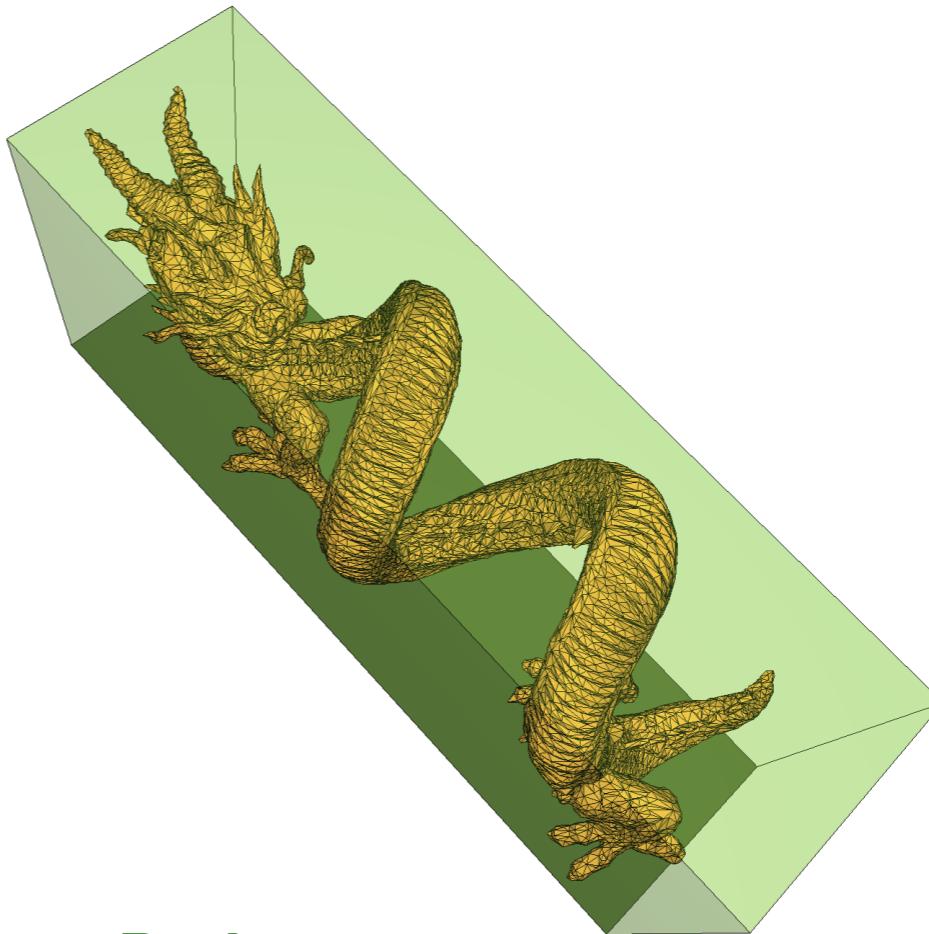


## Pro's

- Fàcil de calcular
- Cost de la intersecció amb el raig molt baix

## Con's

- Es pot perdre molts espais buits
- No cobreix bé la figura



## Pro's

- Adjusta bé la figura
- Menys rajos a testejar

## Con's

- Cost alt de construcció (PCA)
- Cost d'intersecció amb el raig més elevat