

# RECA

Réalisation d'un moteur de rendu 3D  
temps réel réaliste en WebGL

Clément Ange - Reinders Erwan

Sous la direction de Noura Faraj

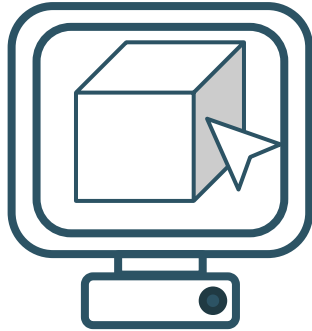


# Sommaire

1. Introduction
2. Détails techniques
3. Moteur de rendu
4. Exemples de shaders
5. Perspectives futures
6. Conclusion



# Introduction





# Détails techniques

Technologies de mise en place d'une solution de rendu 3D



# Technologies utilisées



**Discord**



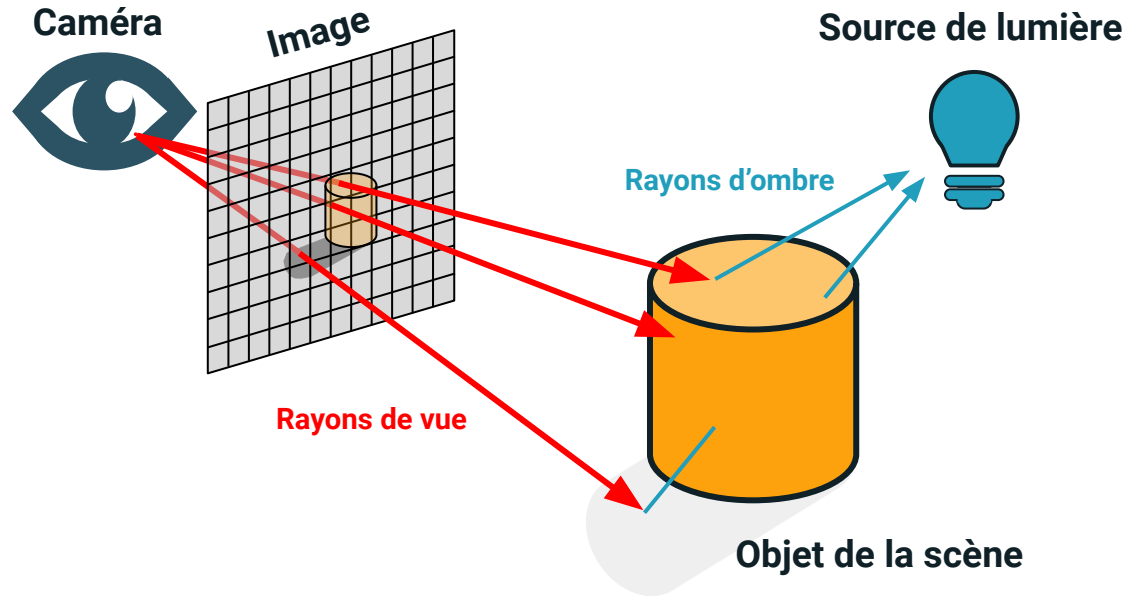
# Spécificités du langage

```
[] == [];  
false  
[] == ![];  
true  
true == [];  
false  
true == ![];  
false  
false == [];  
true  
false == ![];  
true  
!!"false" === !!"true";  
true  
NaN === NaN;  
false  
+!![] / +!![]  
Infinity  
0 == false;  
true
```

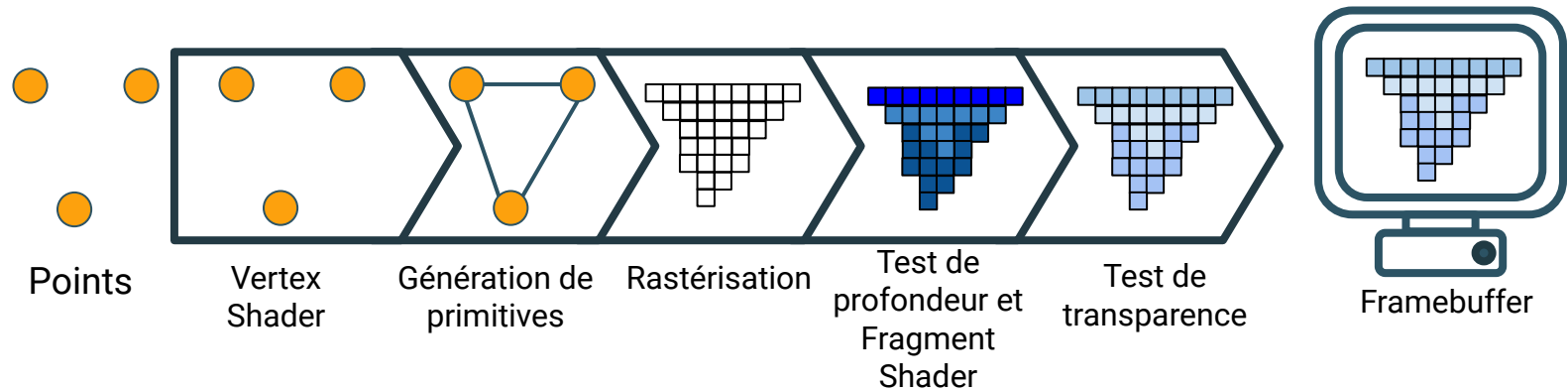
```
"" == false;  
true  
[1, 2, 3] + [4, 5, 6];  
'1,2,34,5,6'  
typeof NaN;  
'number'  
{ } + { }  
'[object Object][object Object]'  
{ } + []  
0  
[] + { }  
'[object Object]'  
(1).__proto__;  
▶ Number {0, constructor: f, toExponential: f, toFixed: f, toPrecision: f, ...}  
(1).__proto__.__proto__;  
▶ {constructor: f, __defineGetter__: f, __defineSetter__: f, hasOwnProperty: f, __lookupGetter__: f, ...}  
(1).__proto__.__proto__.__proto__;  
null  
[...'...']  
▶ (3) [ '.', ' ', ' ']
```

<https://github.com/denysdovhan/wtfjs>

# Comment produire une image à partir d'une scène 3D ?



# Comment produire une image à partir d'une scène 3D ?





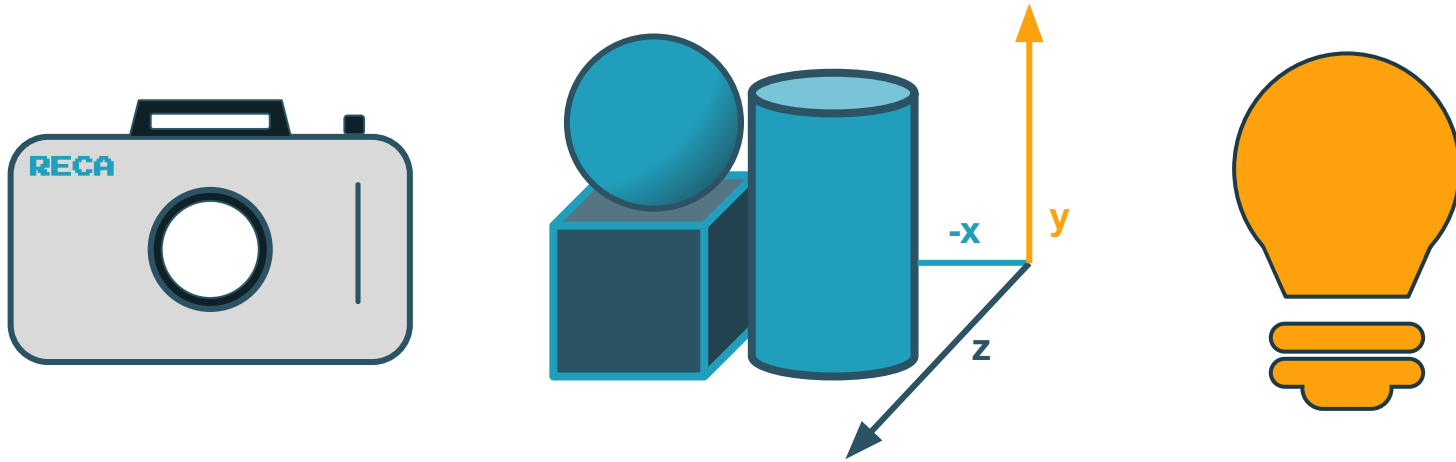


# Moteur de rendu

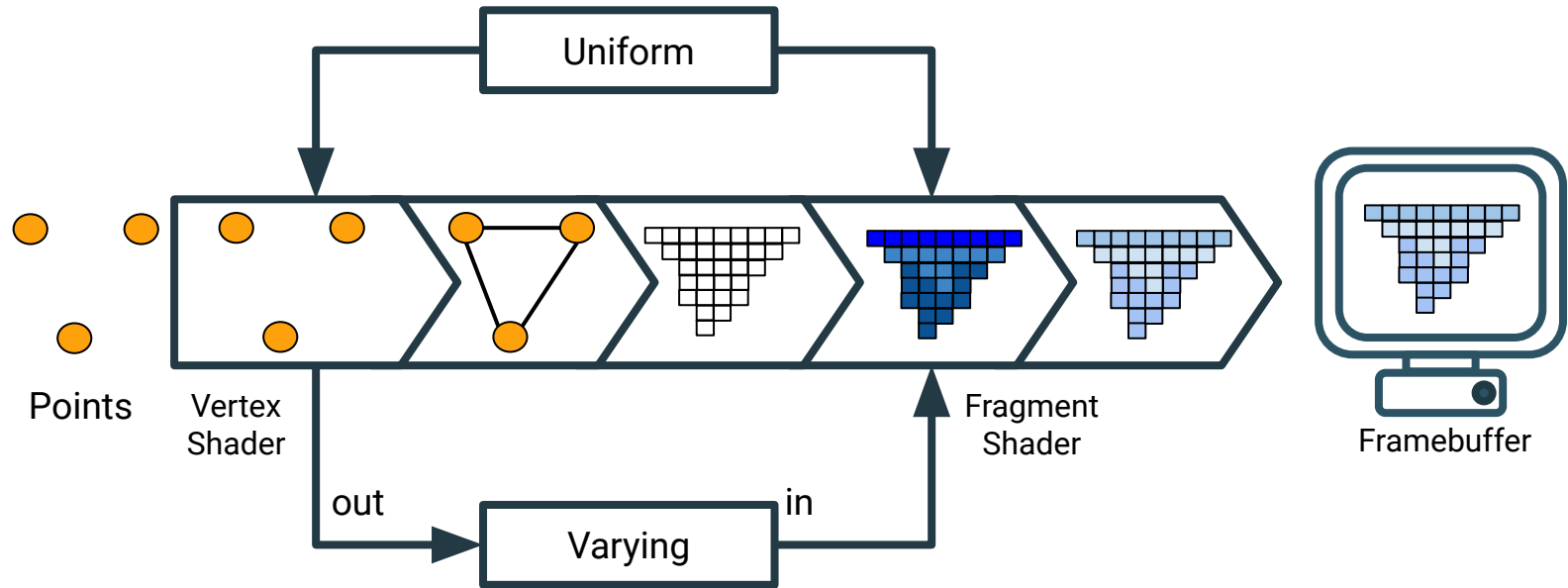
Composants du rendu informatique



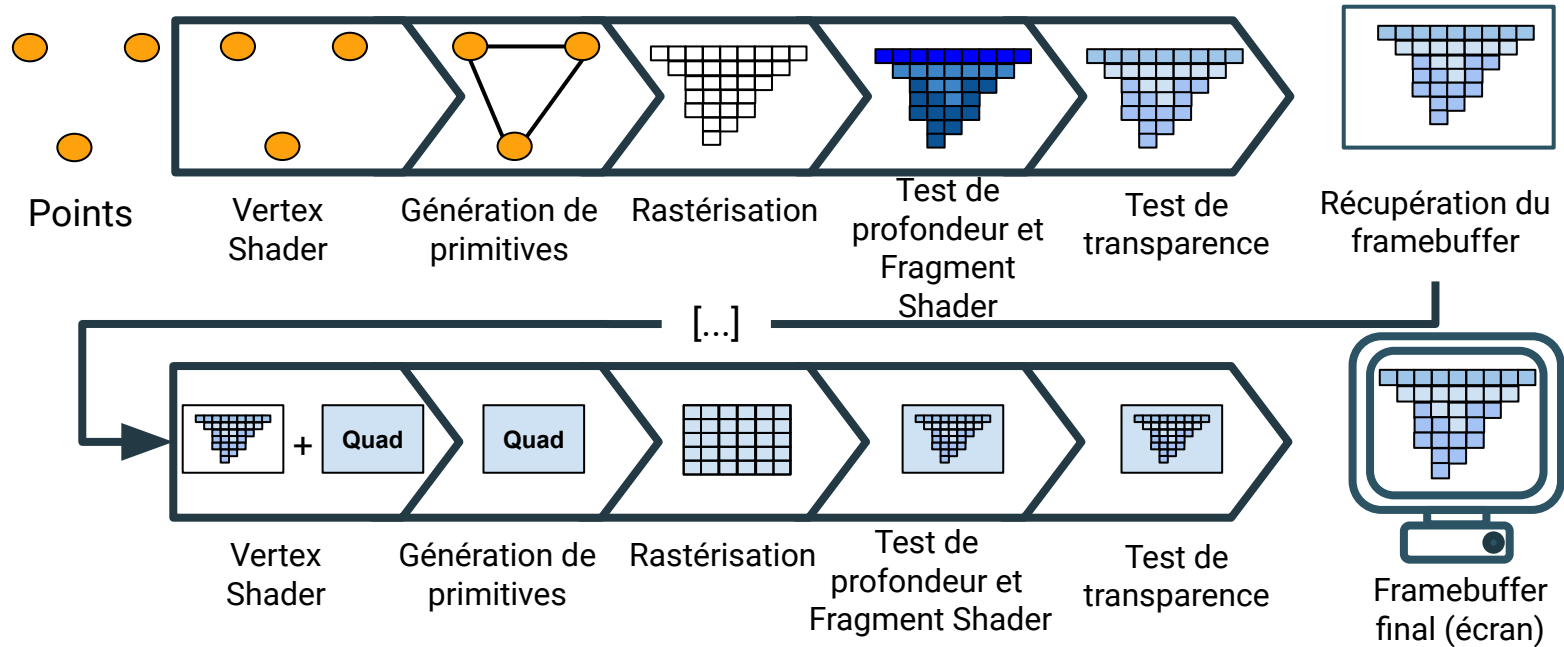
# Composants d'un rendu graphique



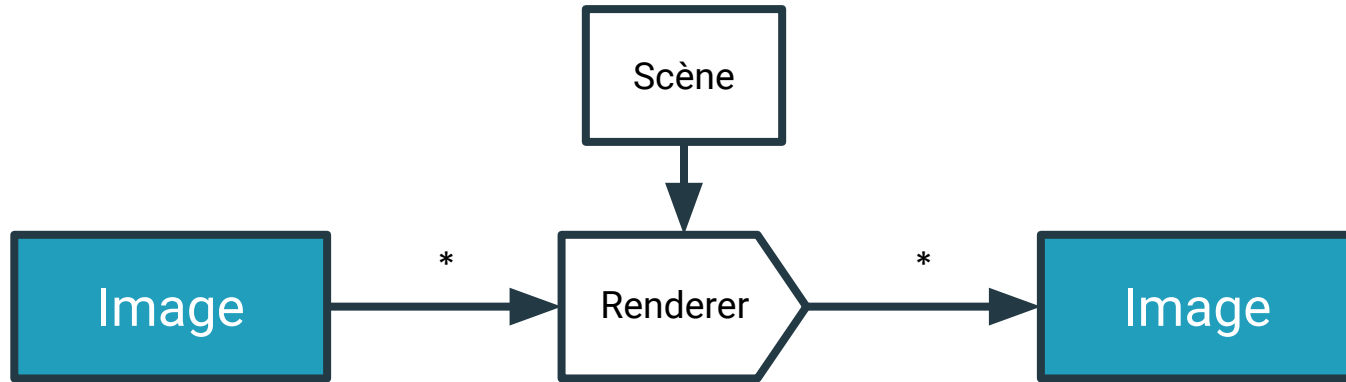
# Encapsulation : shader



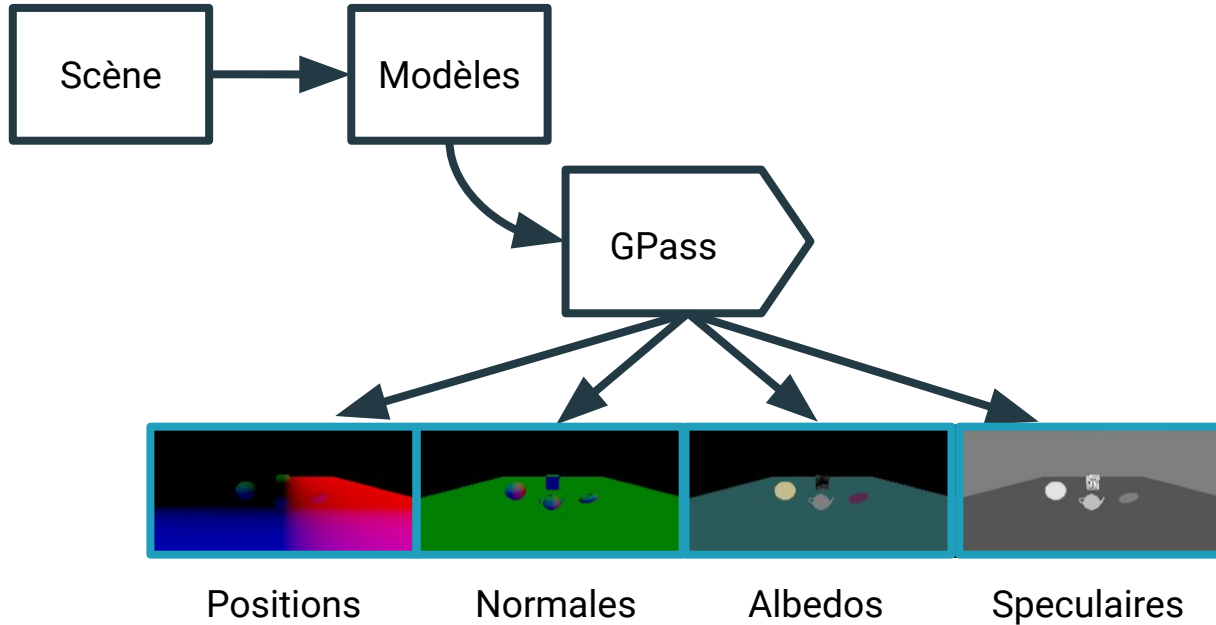
# Encapsulation : shader



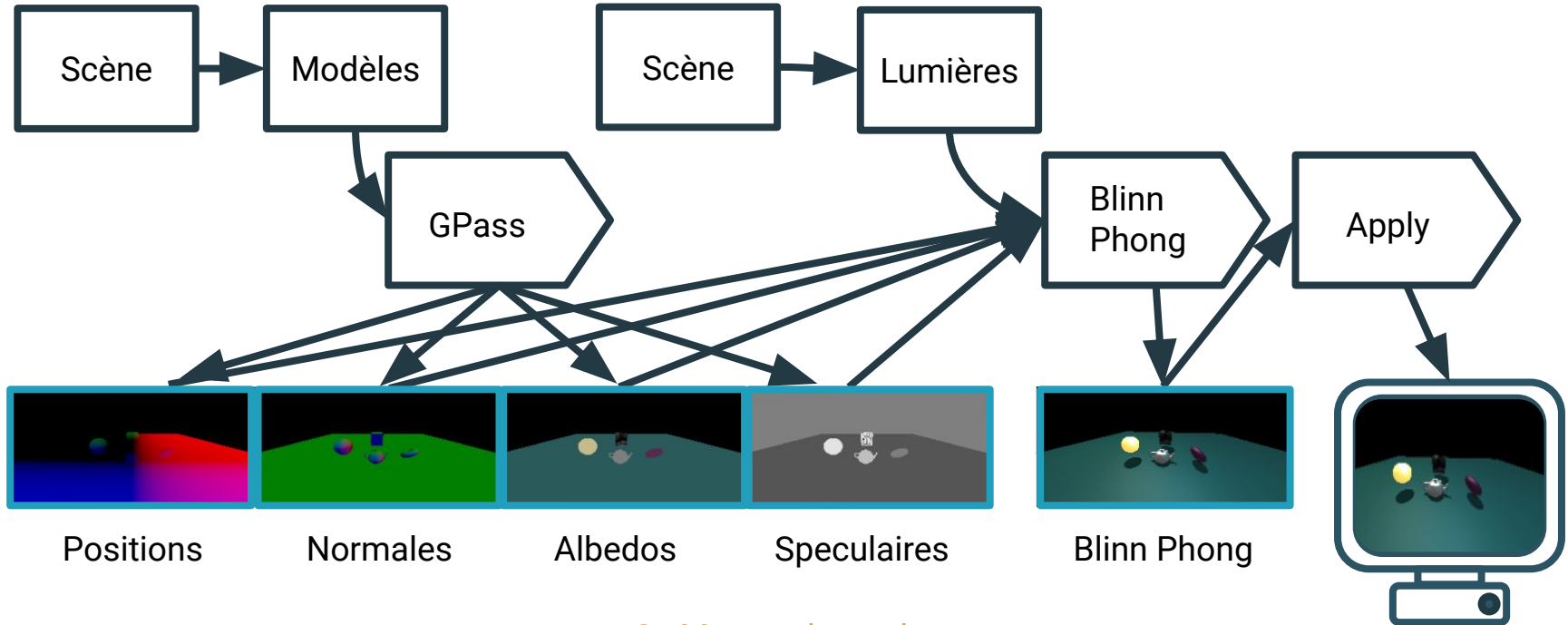
# Encapsulation : renderer



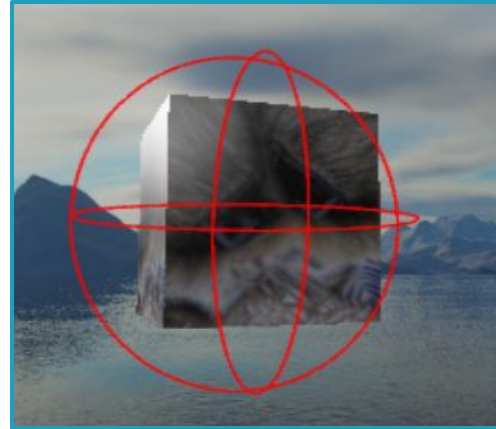
# Encapsulation : renderer



# Encapsulation : pipeline

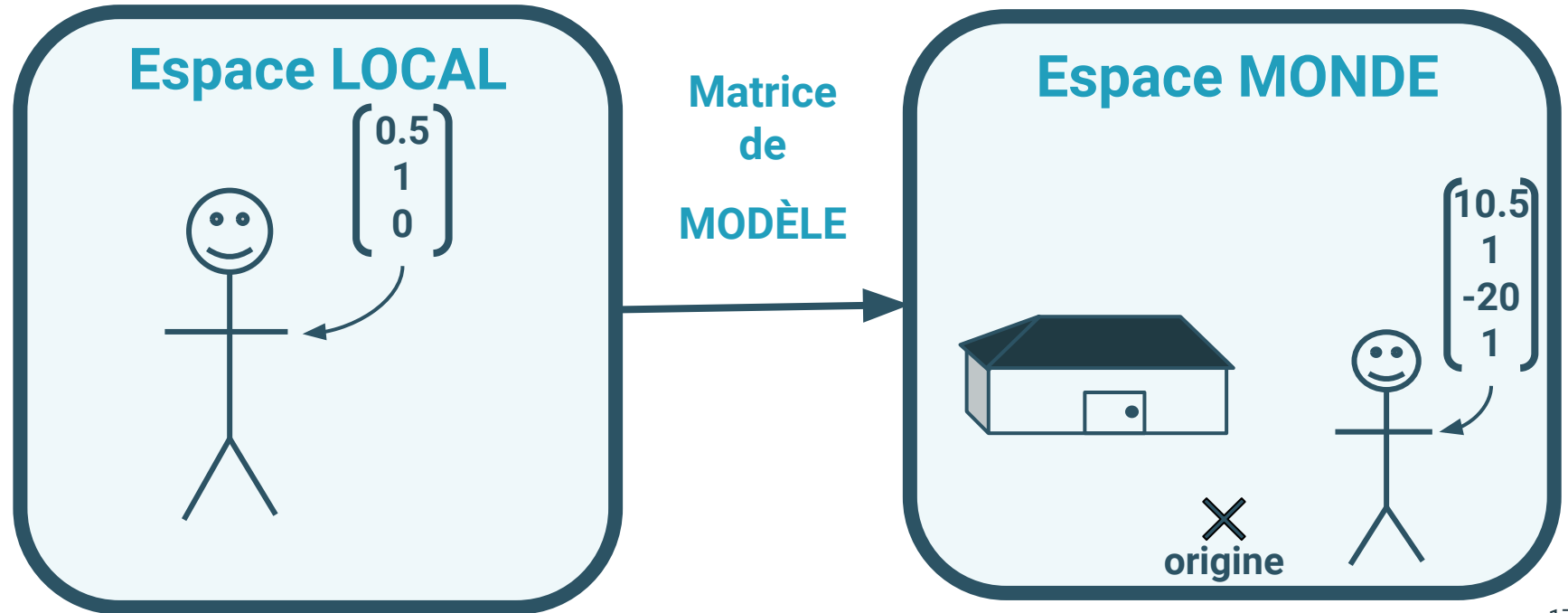


# Interface

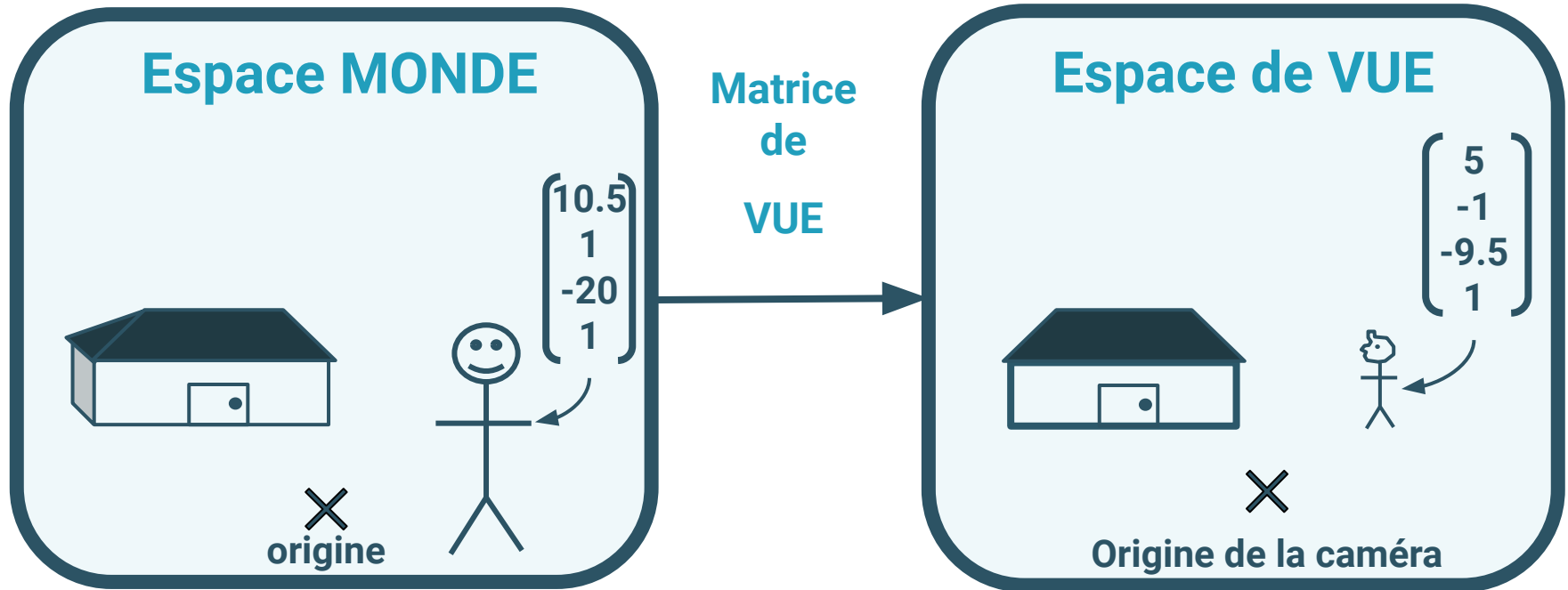




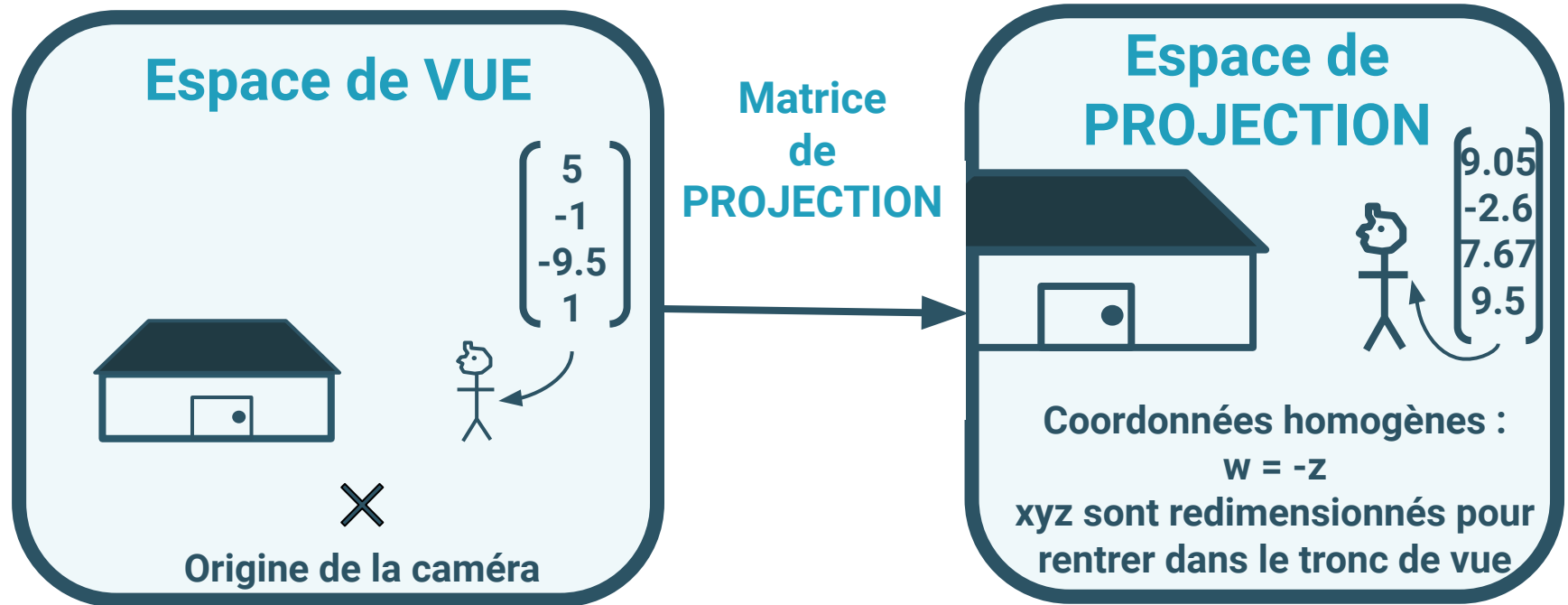
# Sélectionner des objets de la scène



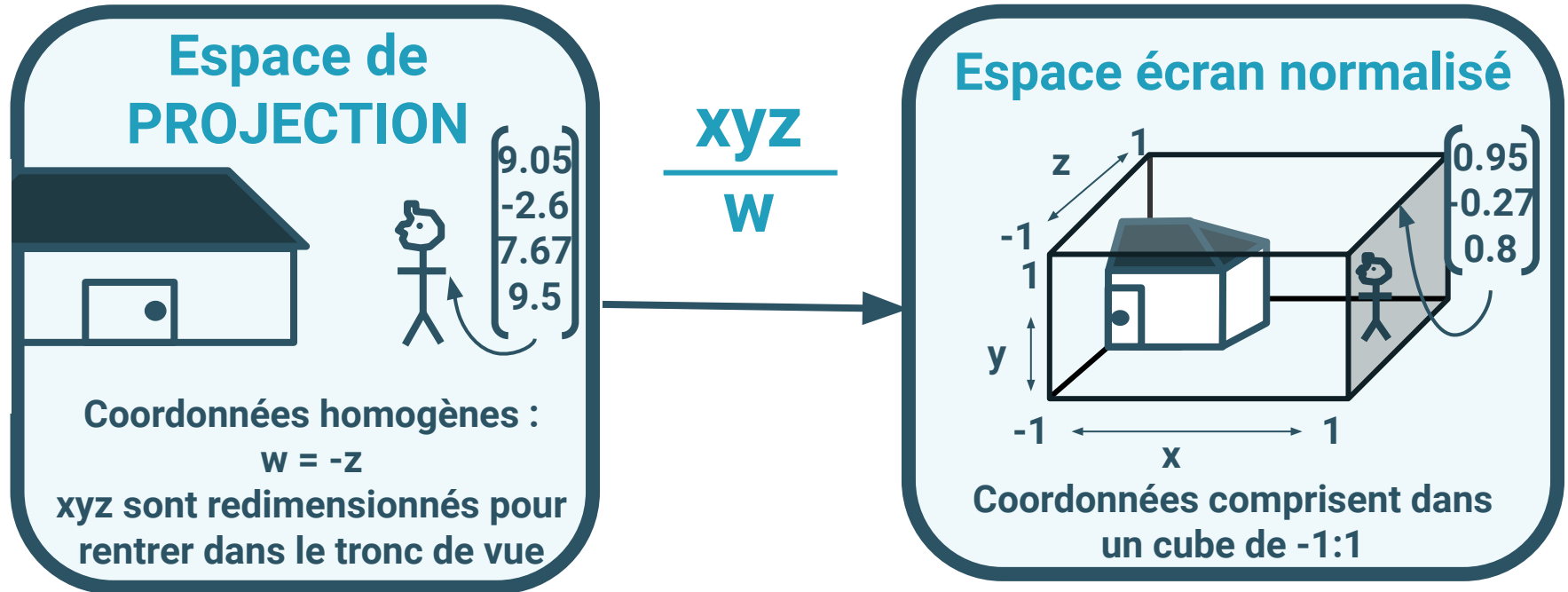
# Sélectionner des objets de la scène



# Sélectionner des objets de la scène

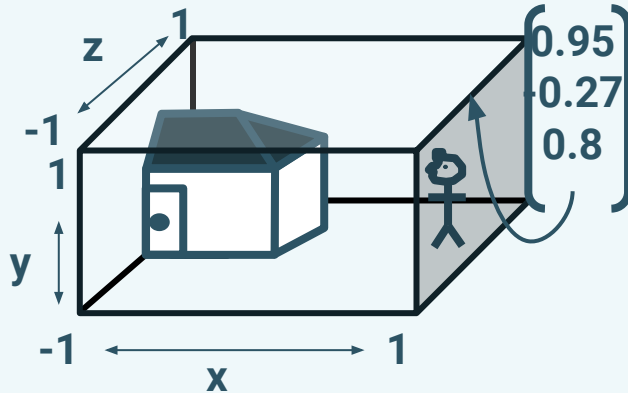


# Sélectionner des objets de la scène



# Sélectionner des objets de la scène

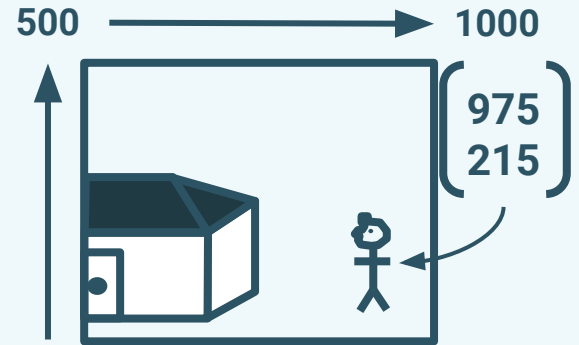
## Espace écran normalisé



Coordonnées comprises dans un cube de -1:1

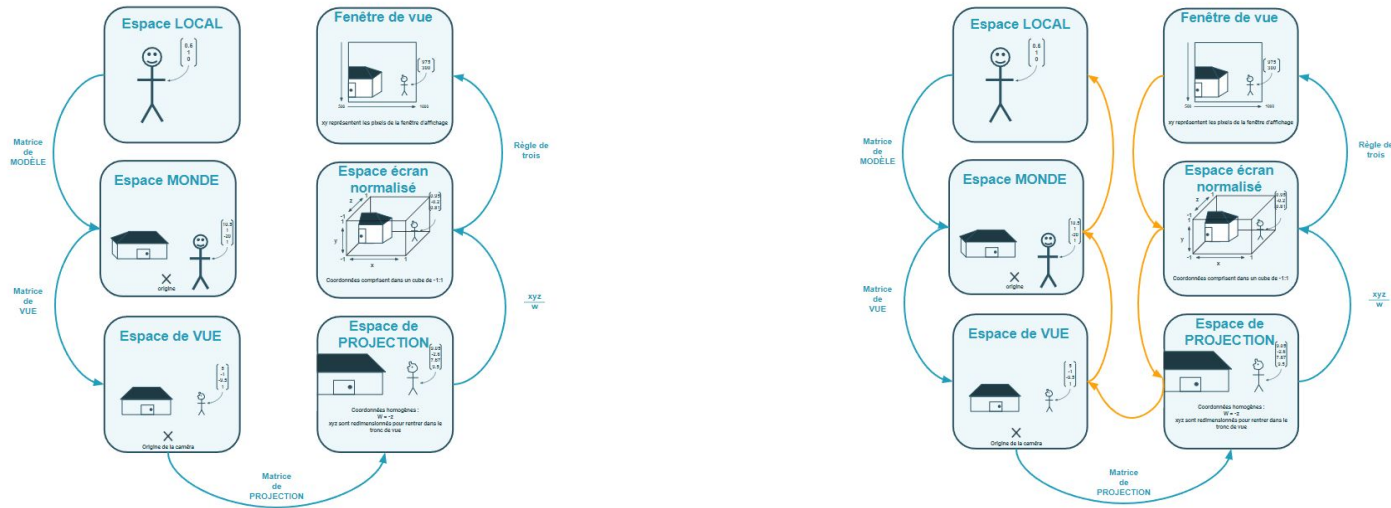
Règle de trois

## Fenêtre de vue



xy représentent les pixels de la fenêtre d'affichage

# Sélectionner des objets de la scène





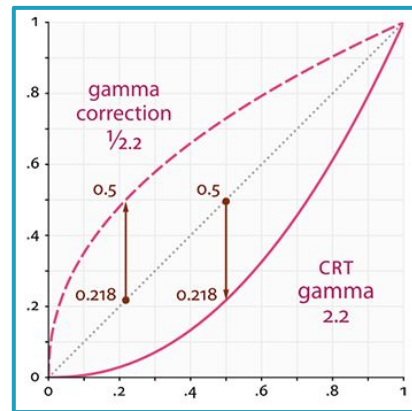
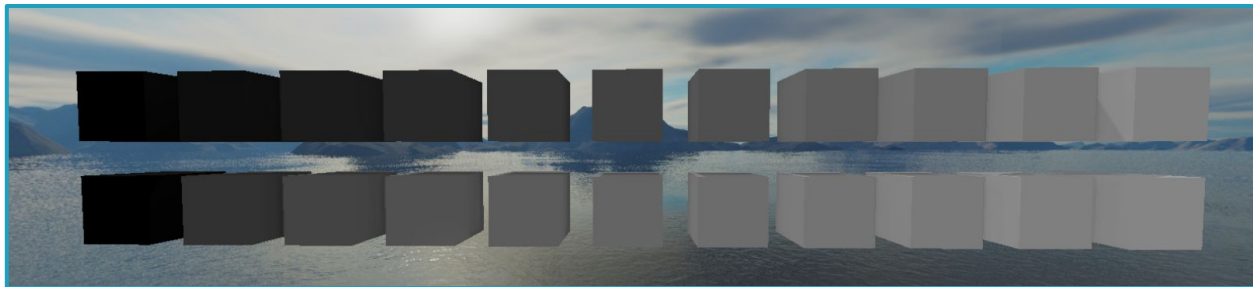
**DÉMONSTRATION**



# **Exemples de shaders**

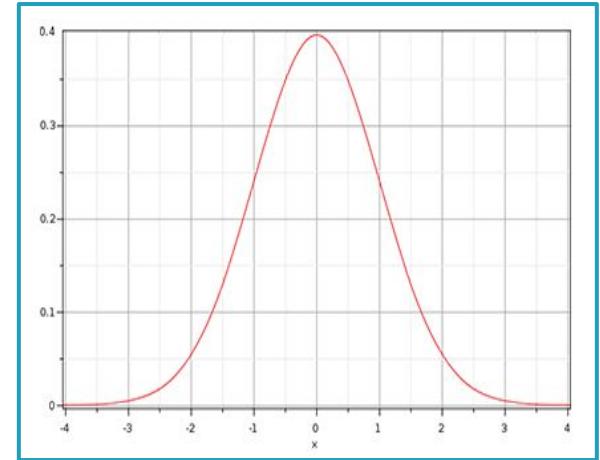
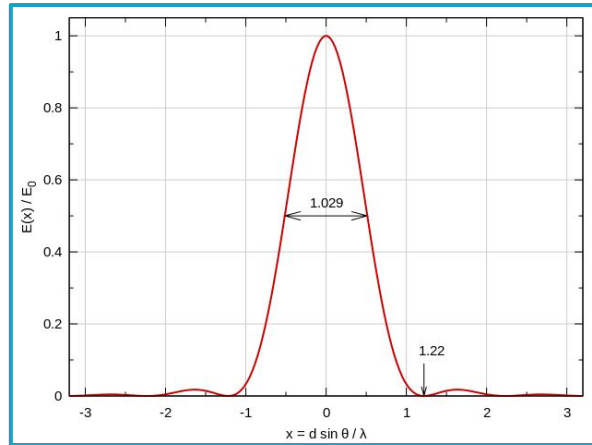
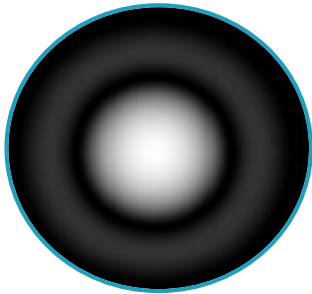


# Gamma correction

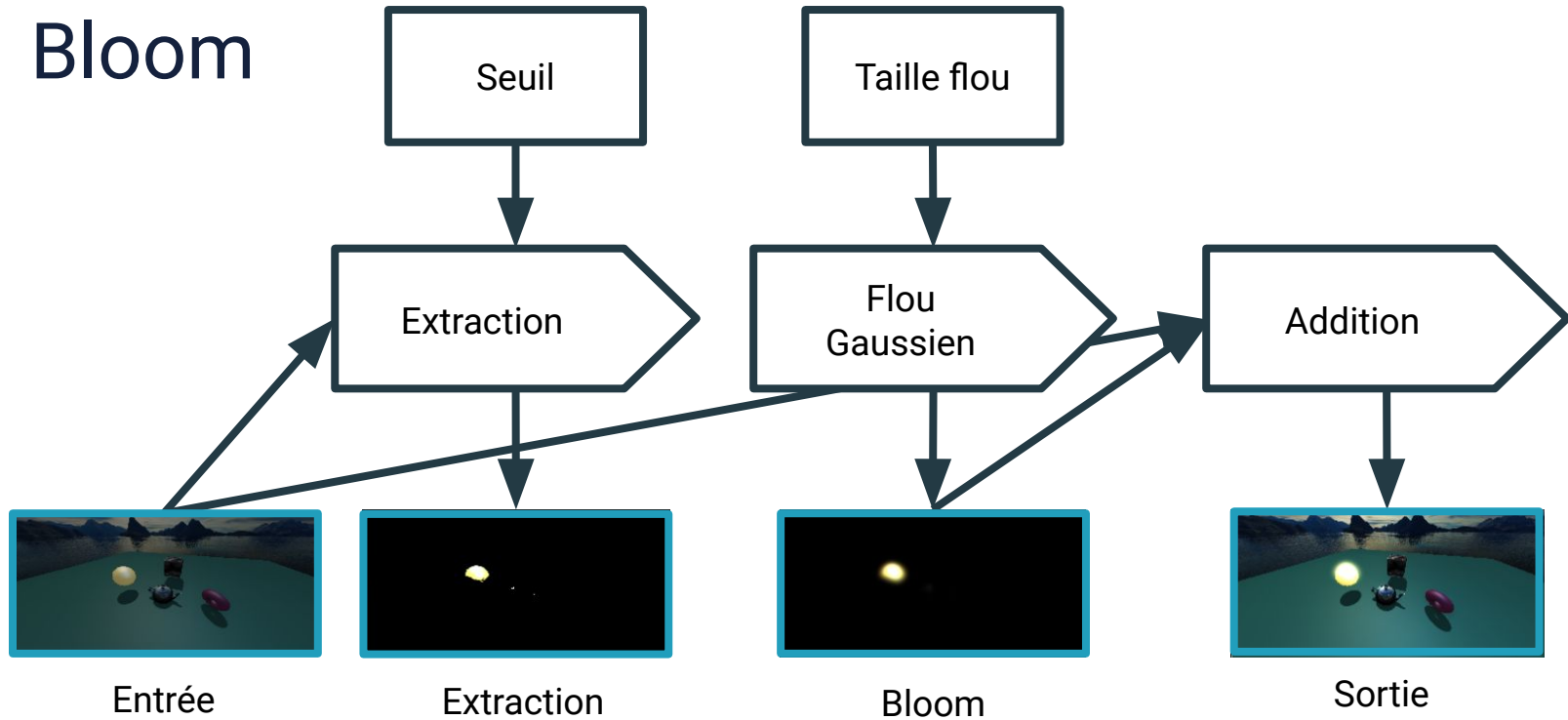


# Bloom

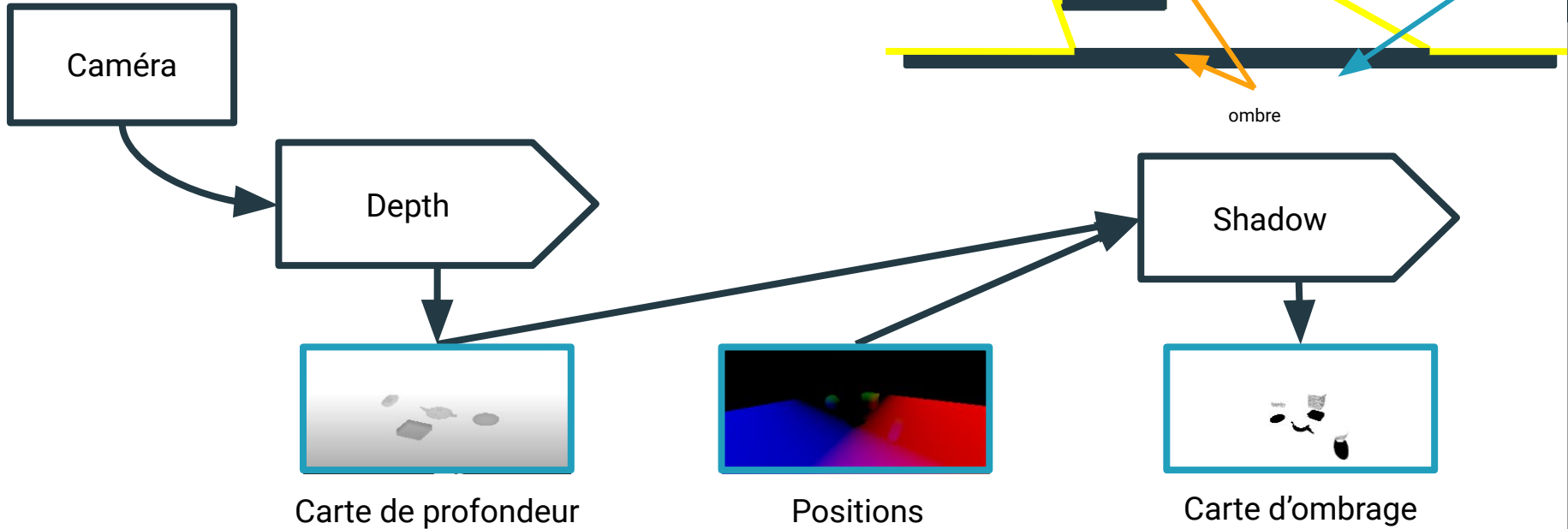
Tâche d'Airy



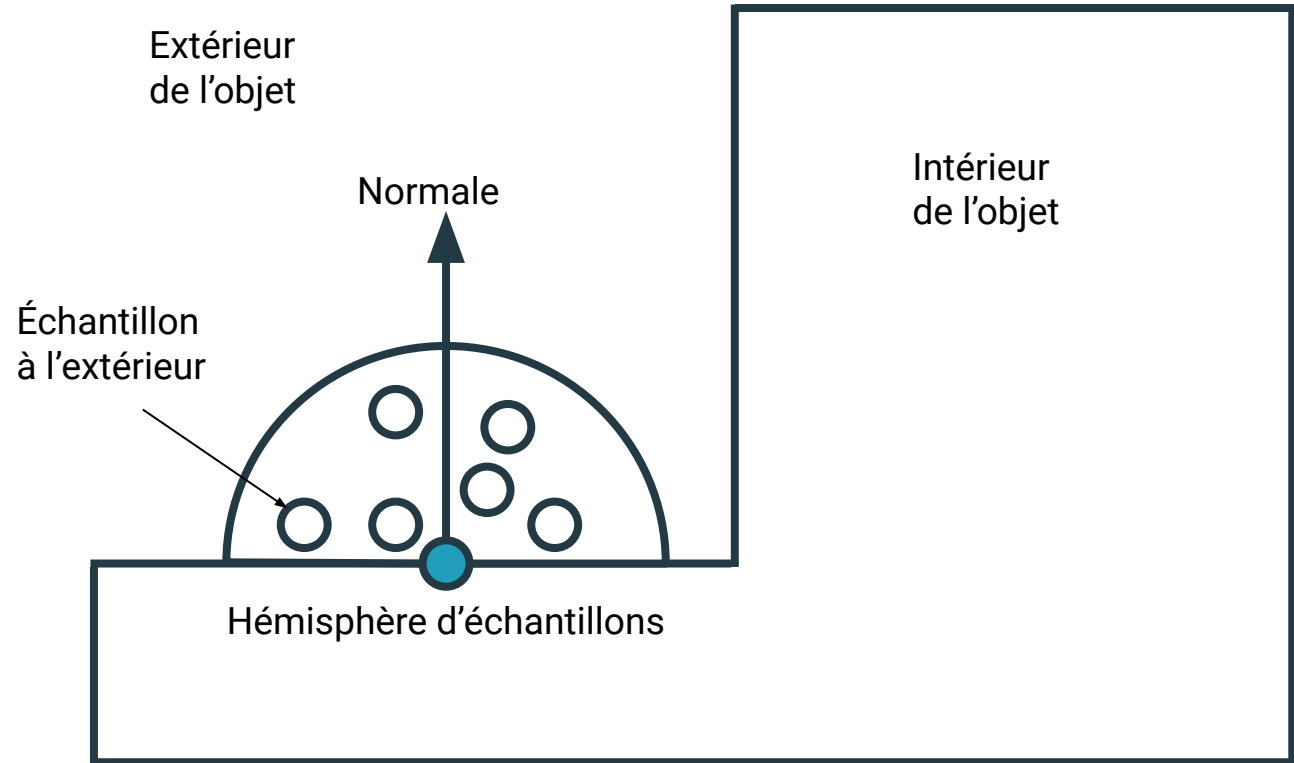
# Bloom



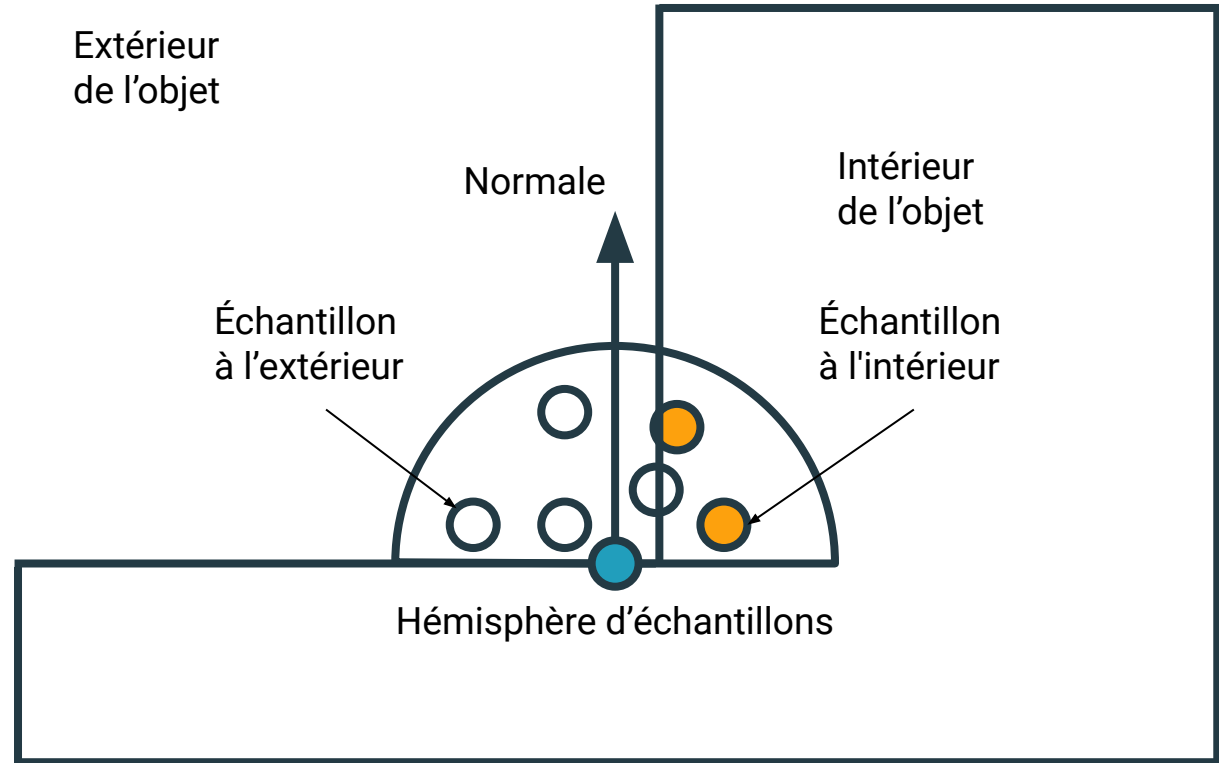
# Shadow mapping



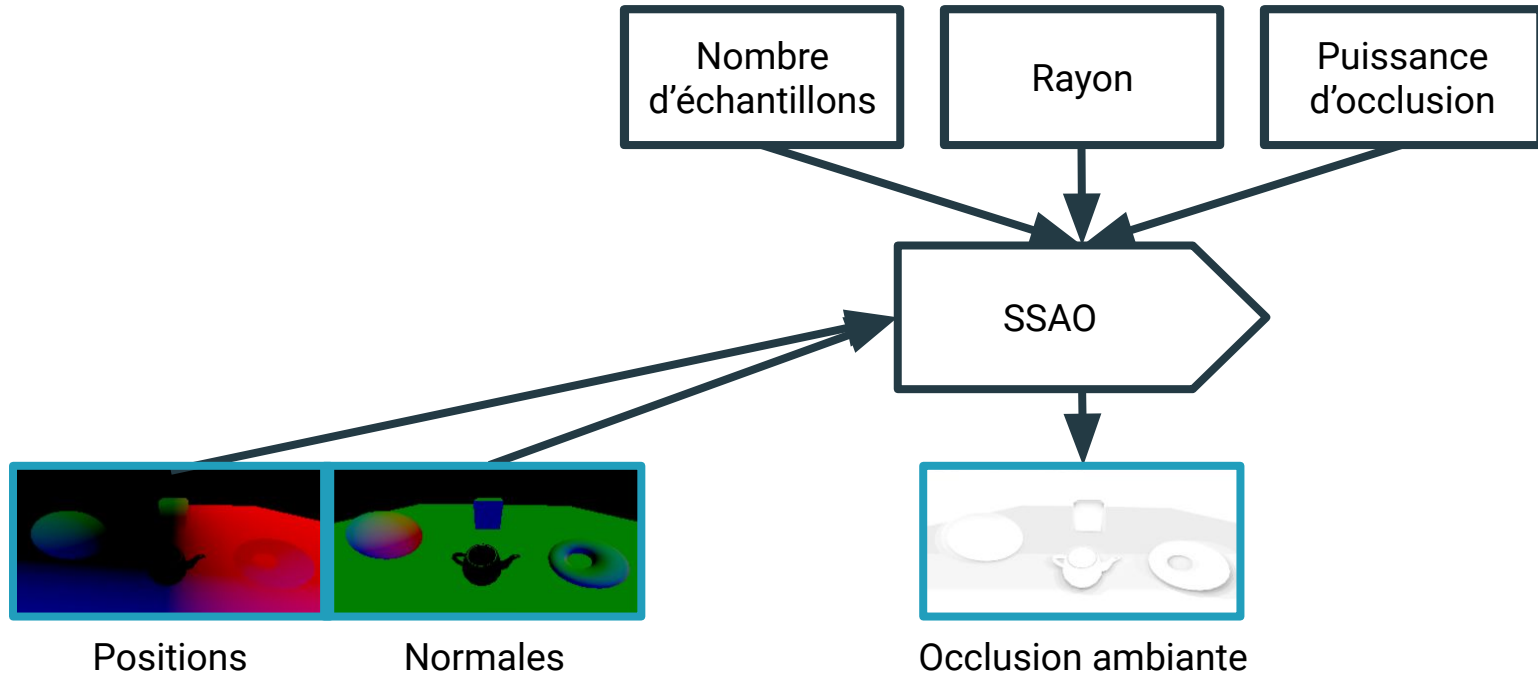
# SSAO



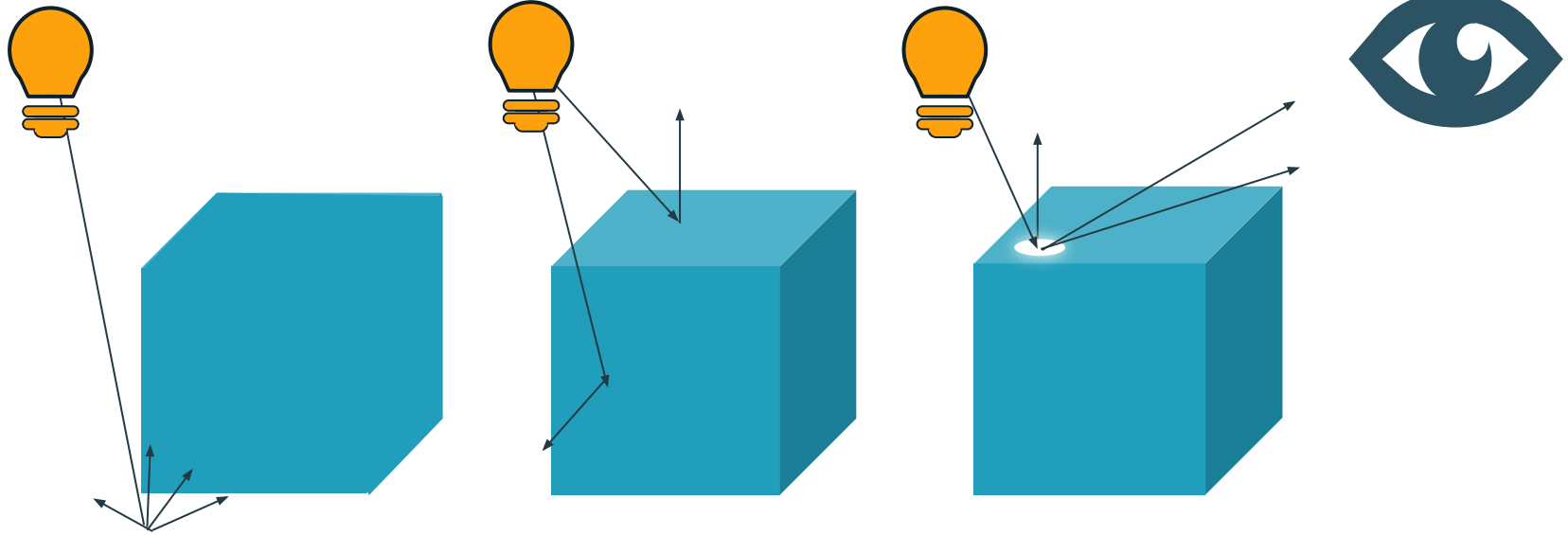
# SSAO



# SSAO

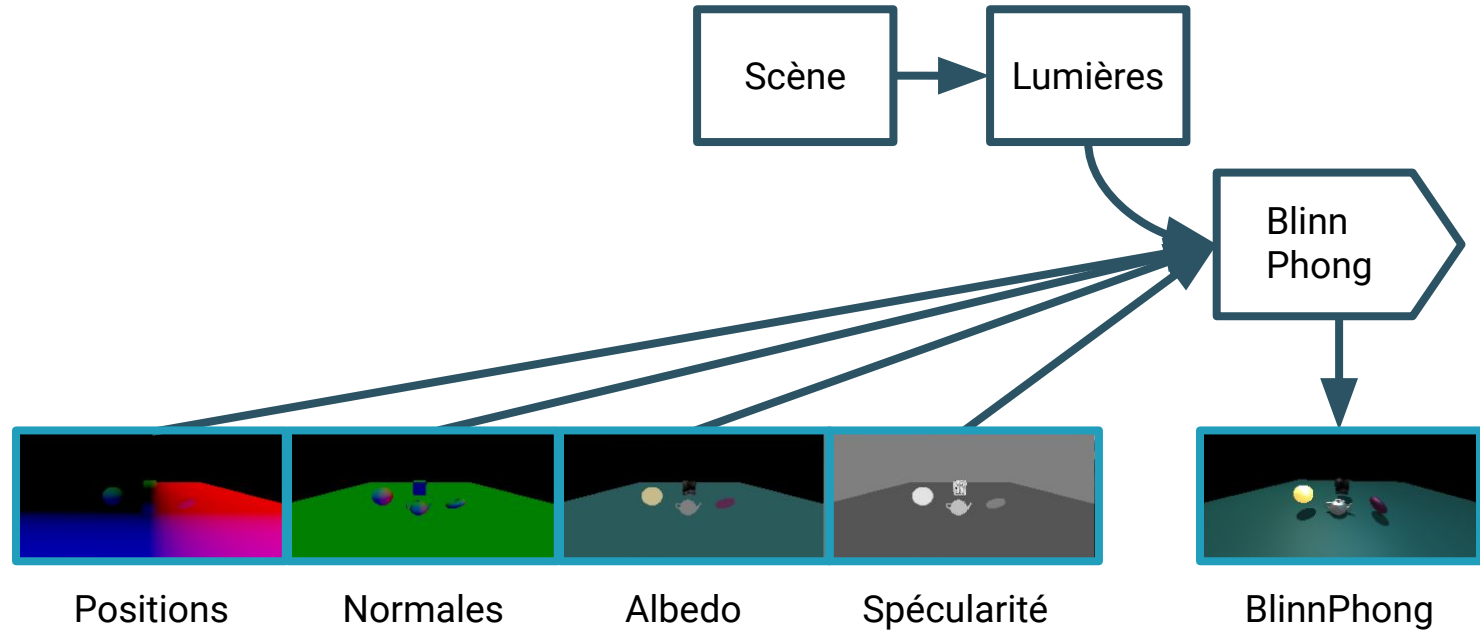
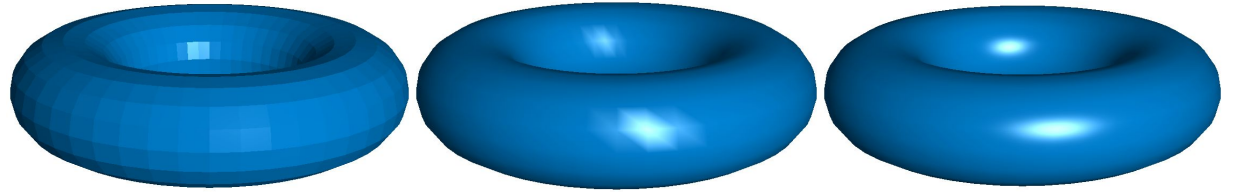


# Blinn Phong





# Blinn Phong



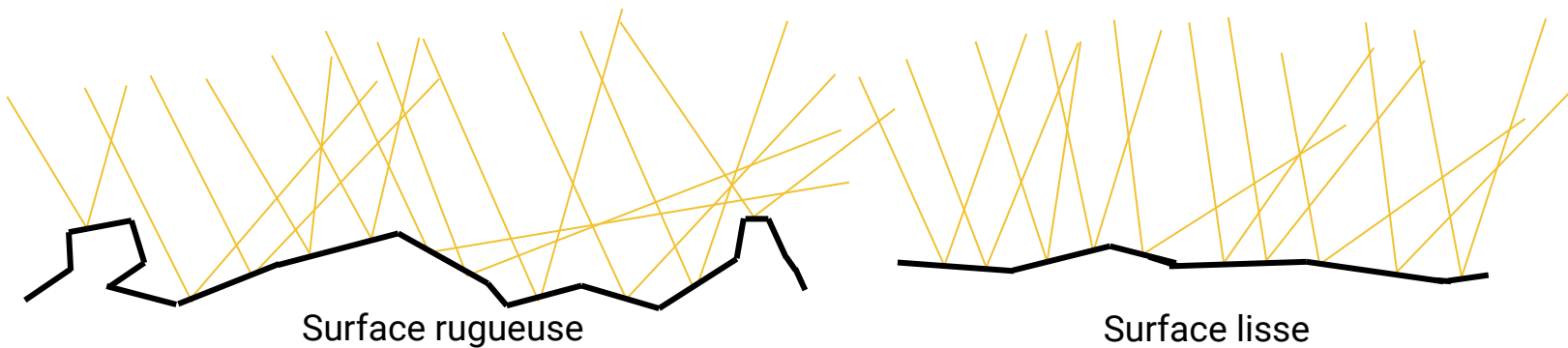
# PBR

1. Microfacettes
2. Conservation énergétique
3. BRDF physique

# PBR

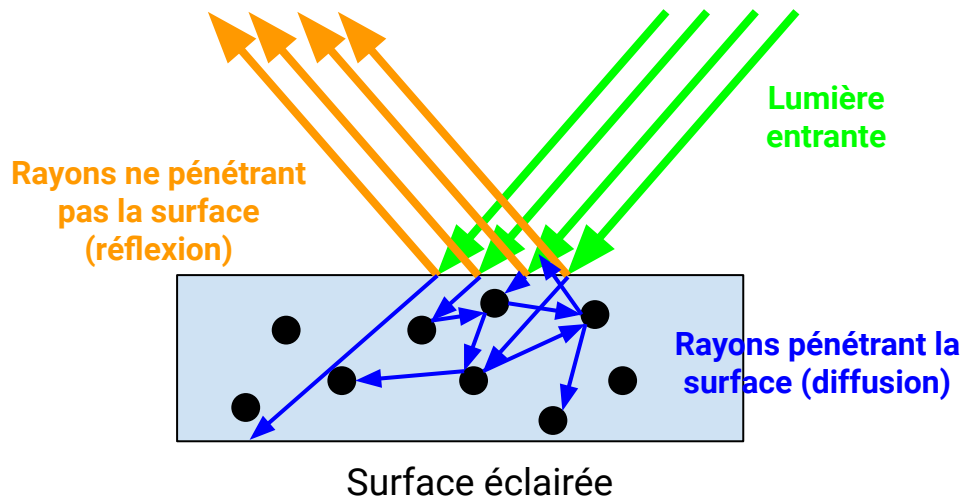
## Microfacettes

$$h = \frac{l+v}{||l+v||}$$



# PBR

## Conservation de l'énergie



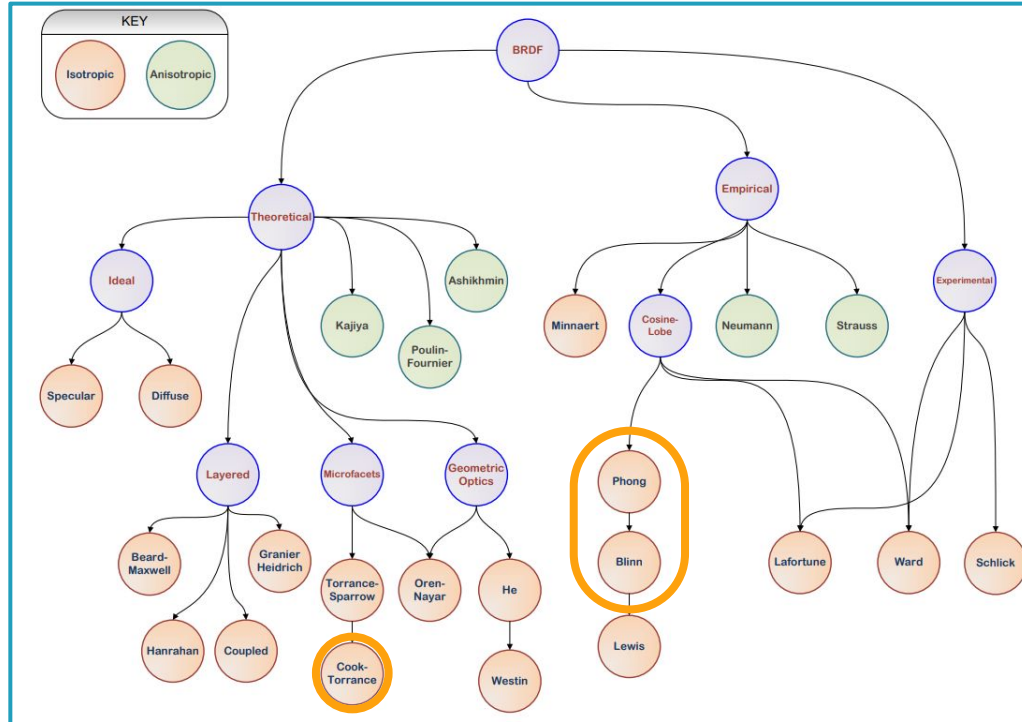
# PBR

## BRDF basée physique

$$L_o(x, w_o, \lambda, t) = L_e(x, w_o, \lambda, t) + \int_{\Omega} f_r(x, w_i, w_o, \lambda, t) L_i(x, w_i, \lambda, t) (w_i \cdot n) dw_i$$

$$IntensiteLumineuseSortante(x, w_o, \lambda, t) = \int_{\Omega} BRDF(x, w_i, w_o, \lambda, t) radiance(x, w_i, \lambda, t) (w_i \cdot n) dw_i$$

# PBR



[https://digibug.ugr.es/bitstream/handle/10481/19751/rmontes\\_LSI-2012-001TR.pdf](https://digibug.ugr.es/bitstream/handle/10481/19751/rmontes_LSI-2012-001TR.pdf)

# PBR

## BRDF basée physique

$$BRDF = kd * flambert + ks * fcook - torrance$$

$$flambert = \frac{albedo}{\pi}$$

$$fcook - torrance = \frac{DFG}{4(w0.n)(wi.n)}$$

Modèle spéculaire des microfacettes de la BRDF



# PBR

## BRDF basée physique

- **D** : Distribution des normales (NDF)
- **F** : équation de Fresnel décrivant le rapport de la réflexion de la surface à différents angles
- **G** : fonction de géométrie décrivant la propriété d'auto-ombrage d'un objet



# PBR et propriétés matérielles

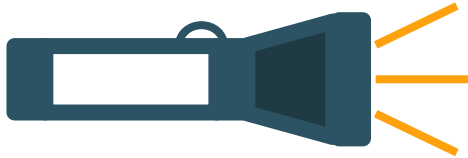
- Rugosité
- Taux de métallisation
- F0
- Occlusion ambiante
- Carte de normales



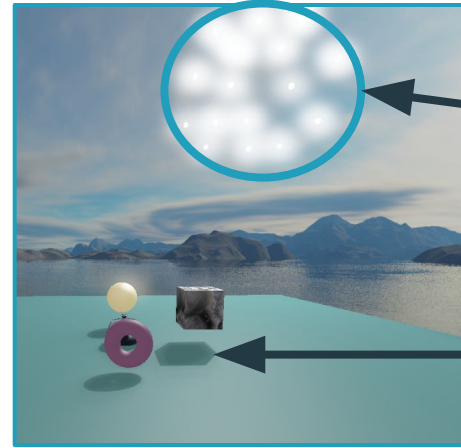
# **Perspectives futures**

# Autres types de lumières

Spot de lumière

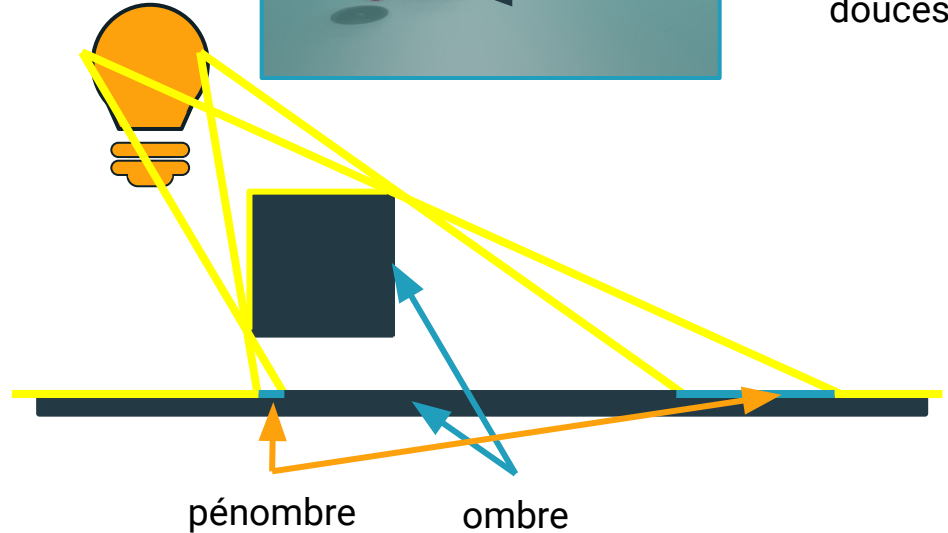


Néon

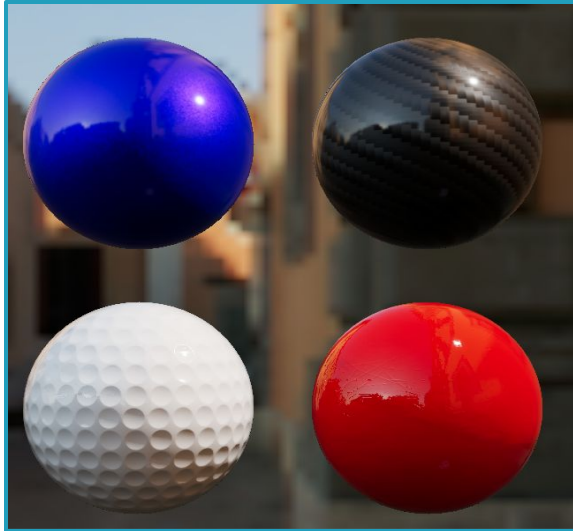


Lumière étendue

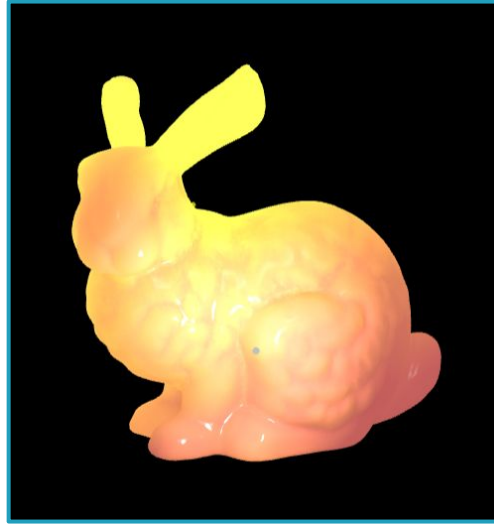
Ombres douces



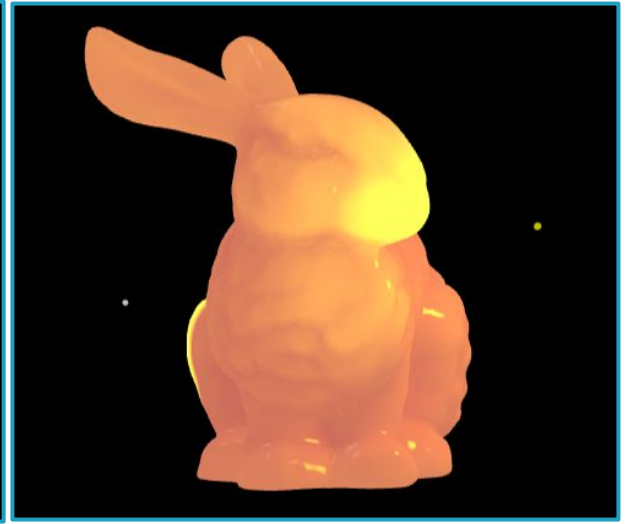
# Plus de PBR



[https://threejs.org/examples/?q=aniso#webgl\\_materials\\_physical\\_clearcoat](https://threejs.org/examples/?q=aniso#webgl_materials_physical_clearcoat)



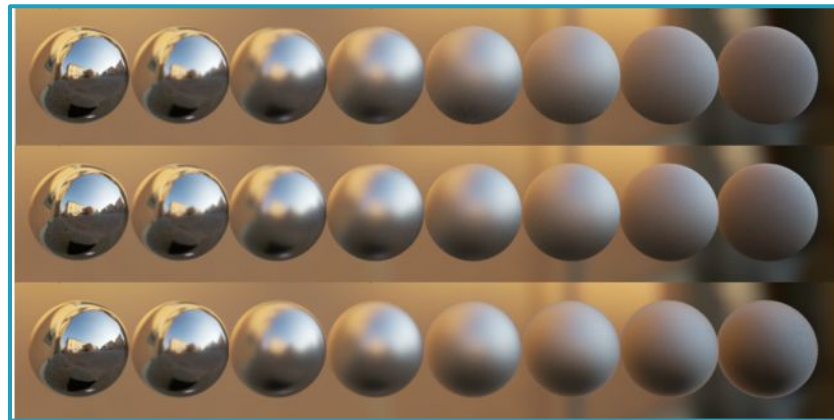
[https://threejs.org/examples/webgl\\_materials\\_subsurface\\_scattering.html](https://threejs.org/examples/webgl_materials_subsurface_scattering.html)



$$G(\mathbf{v}; \mu, \lambda, a) = ae^{\lambda(\mu \cdot \mathbf{v} - 1)}$$

<https://mynameismjp.wordpress.com/2016/10/09/sg-series-part-6-step-into-the-baking-lab/>

# PBR et éclairage



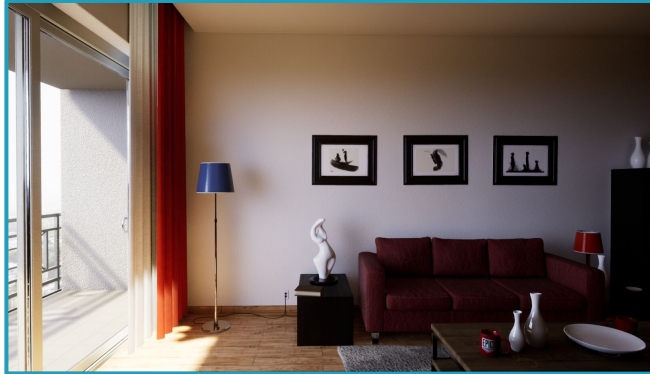
[https://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013\\_pbs\\_epic\\_notes\\_v2.pdf](https://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf)

$$Lo(p, wo) = \int_{\Omega} (kd \frac{c}{\pi}) Li(p, wi)(wi.n) dwi + \int_{\Omega} (ks \frac{DFG}{4(w0.n)(wi.n)}) Li(p, wi)(wi.n) dwi$$

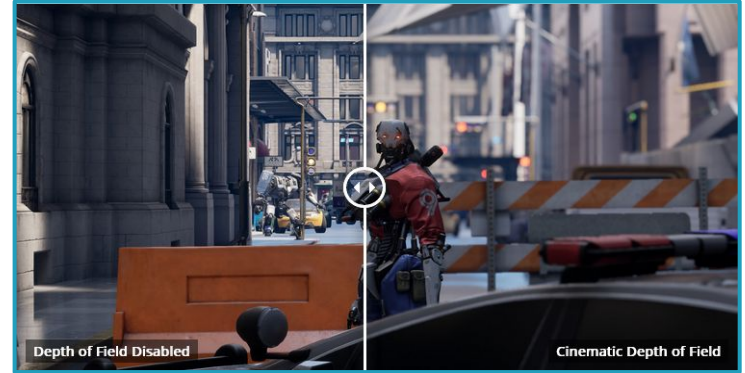
$$\int_{\Omega} Li(p, wi) dwi * \int_{\Omega} BRDF(p, wi, wo)(wi.n) dwi$$

5 - Perspectives futures

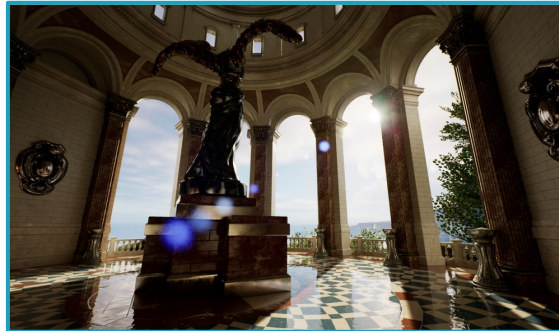
# Autres méthodes



Illumination globale (Unreal engine 5)



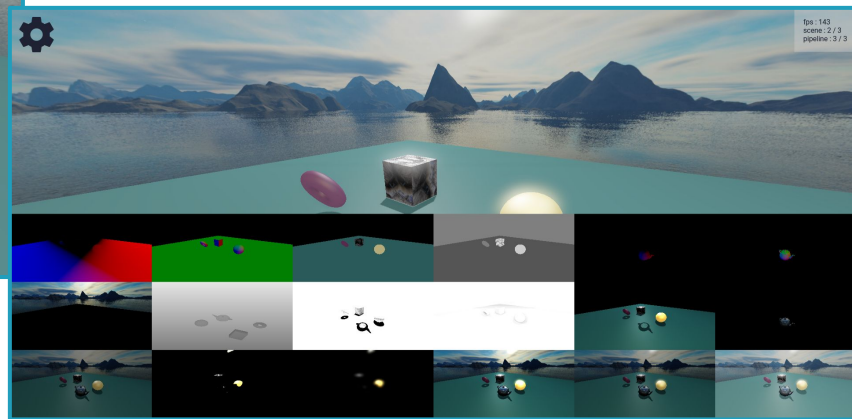
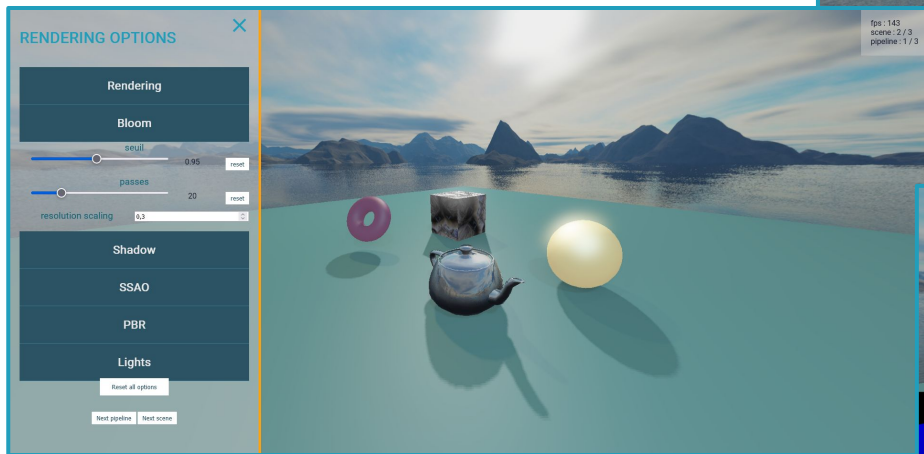
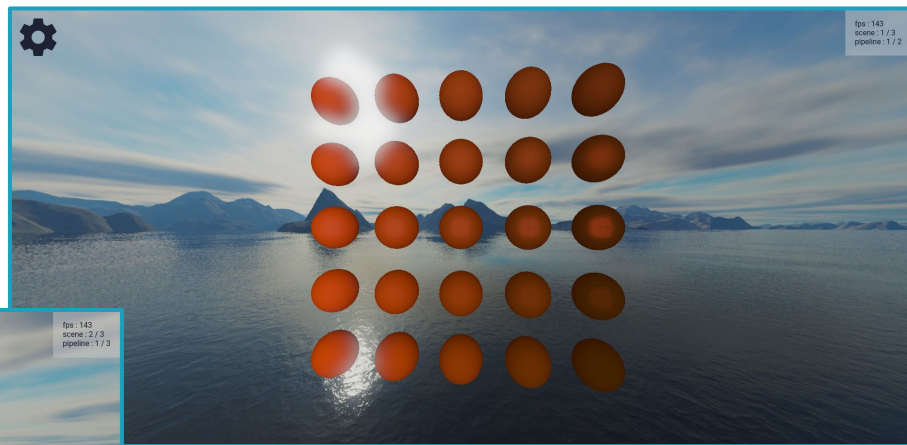
Profondeur de champ (Unreal engine 5)



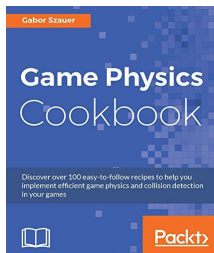
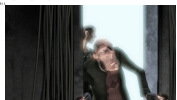
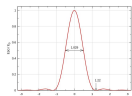
Lens flare (Unreal engine 5)

<https://docs.unrealengine.com/5.0/en-US/designing-visuals-rendering-and-graphics-with-unreal-engine/>

# Conclusion







By <sup>™</sup>/©Khronos Group - Vectorised by Vulphere from <https://www.khronos.org/legal/trademarks/>, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=52861065>

Par Discord — [https://commons.wikimedia.org/w/index.php?title=File:Discord\\_logo\\_\(2021\).svg&lang=fr](https://commons.wikimedia.org/w/index.php?title=File:Discord_logo_(2021).svg&lang=fr), marque déposée, <https://fr.wikipedia.org/w/index.php?curid=14160058>

Par GitHub — <https://github.com/github/octicons>, MIT, <https://commons.wikimedia.org/w/index.php?curid=33440579>

Par Fffred sur Wikipédia français — Transféré de fr.wikipedia à Commons par Bloody-libu utilisant CommonsHelper., Domaine public, <https://commons.wikimedia.org/w/index.php?curid=16057464>

Par Edgar Bonet — Travail personnel, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=44717178>

By Ton Roosendaal et al. - The 1024x576 version of the film., CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=4006098>

Articles opengl par Joey de Vries :

Gamma correction : <https://learnopengl.com/Advanced-Lighting/Gamma-Correction>

Bloom : <https://learnopengl.com/Advanced-Lighting/Bloom>

Shadow mapping : <https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping>

SSAO : <https://learnopengl.com/Advanced-Lighting/SSAO>

Sources PBR :

Filament : <https://google.github.io/filament/Filament.html>

Github de Sébastien Beugnon : <https://github.com/sbeugnon/pbr-cg-2020>

BRDF : [https://digibug.ugr.es/bitstream/handle/10481/19751/rmontes\\_LSI-2012-001TR.pdf](https://digibug.ugr.es/bitstream/handle/10481/19751/rmontes_LSI-2012-001TR.pdf)

Implémentations :

<https://learnopengl.com/PBR/Lighting>

[https://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013\\_pbs\\_epic\\_notes\\_v2.pdf](https://blog.selfshadow.com/publications/s2013-shading-course/karis/s2013_pbs_epic_notes_v2.pdf)

[https://media.disneyanimation.com/uploads/production/publication\\_asset/48/asset/s2012\\_pbs\\_disney\\_brdf\\_notes\\_v3.pdf](https://media.disneyanimation.com/uploads/production/publication_asset/48/asset/s2012_pbs_disney_brdf_notes_v3.pdf)