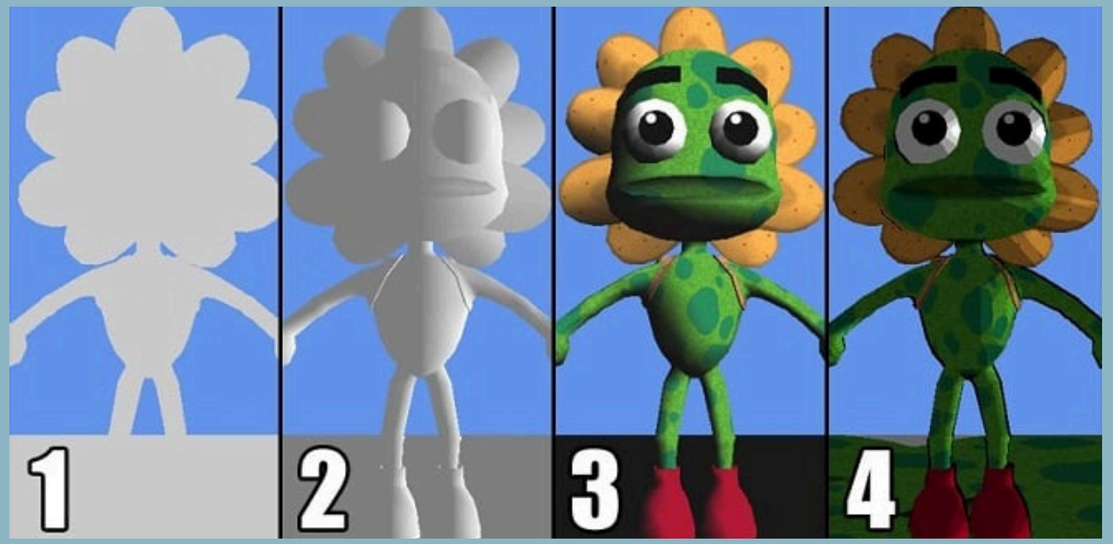
Les shaders

Dans les jeux

Formation CELL par Crizomb

Sommaire
de la présentation

01 *Qu’est-ce qu’un shader ?*

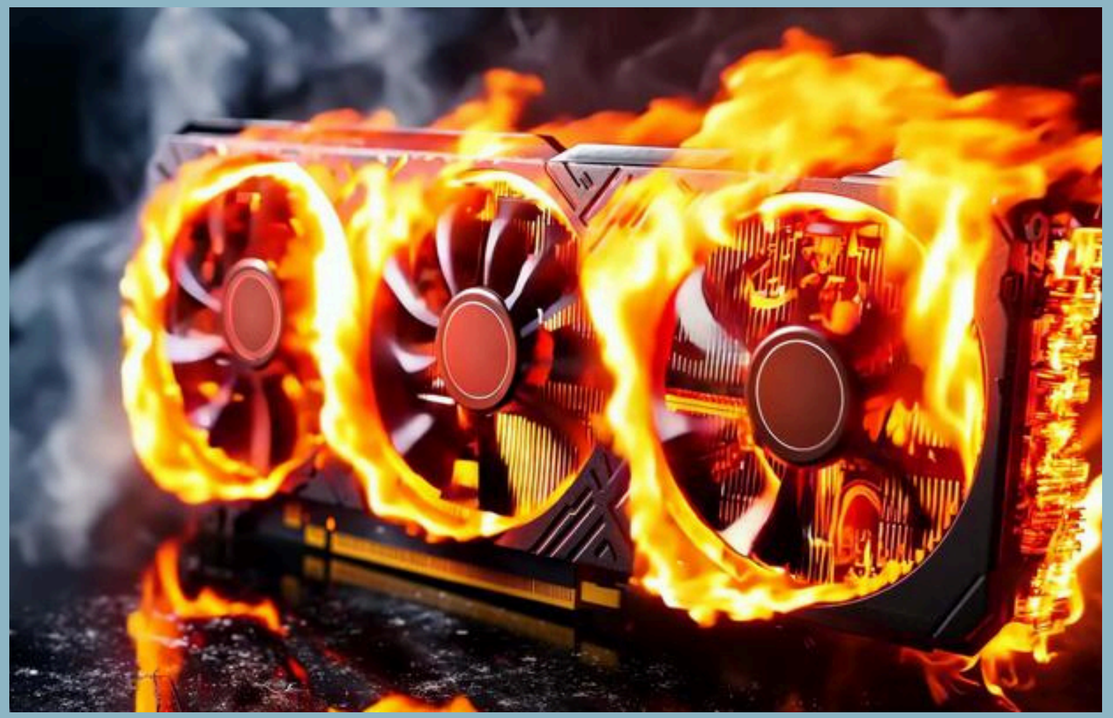


Définition originelle

Programme informatique utilisé pour faire des calculs de lumière et de texture sur des modèles 3D

Définition dans le jeux-vidéo

Programme informatique ultra parallélisé qui tourne sur GPU

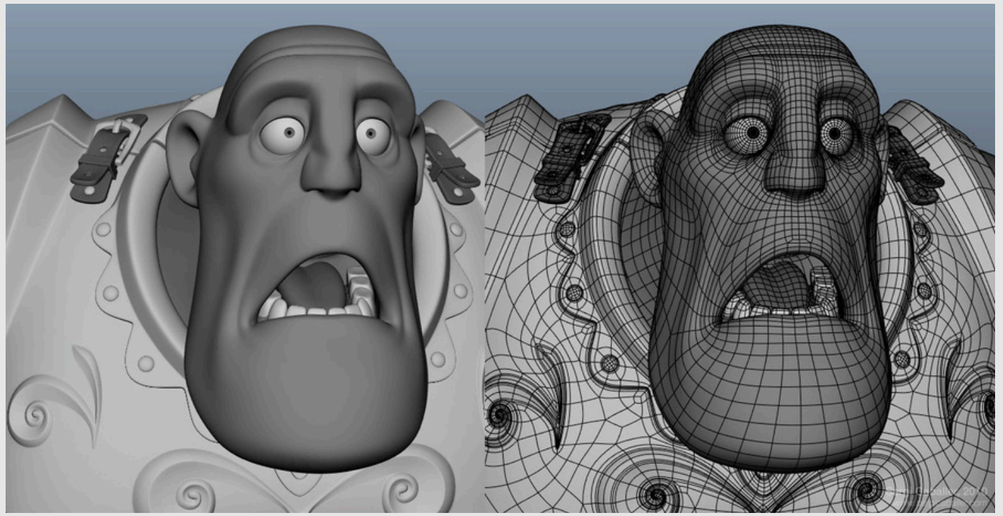


Le GPU (Carte graphique)

- Super fort en calcul parallèle
- Possède des milliers de coeurs
- Parfait pour des images et des modèles 3D avec des millions et de milliers de pixels et de faces

02

Base du shading



Un modèle 3D

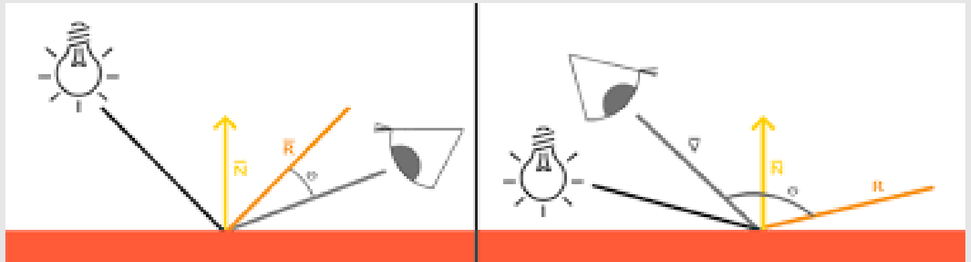
- Des points (vertex) et des faces

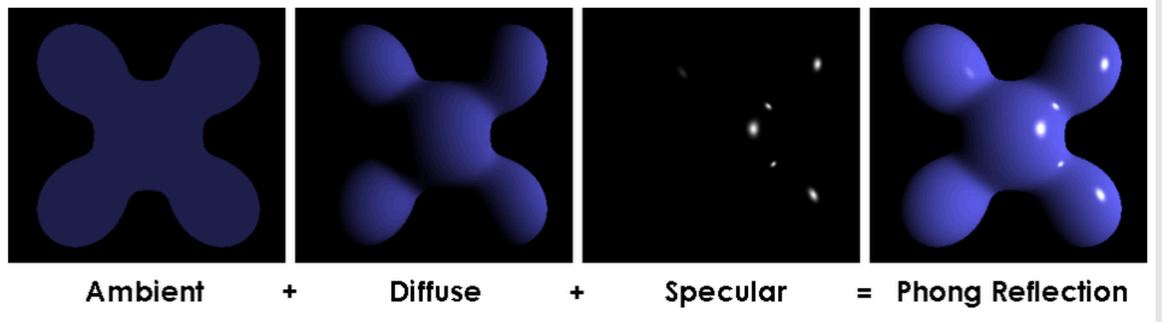
Le shading c’est l’addition de :

- Ambiant (couleur intrinsèque de l’objet)
- Diffuse (Est-ce que la surface reçoit beaucoup de lumière)
- Specular (Réflexion direct de la surface sur nos yeux)

Ce dont on a besoin :

- Position de la lumière
- Position de la caméra (oeil)
- Caractéristiques de l’objet en chaque point (matériau)
  - Couleur
  - Vecteur normal
  - Roughness (tendance à faire du specular)





03 *Les différents types de shaders*

Fragment shader:

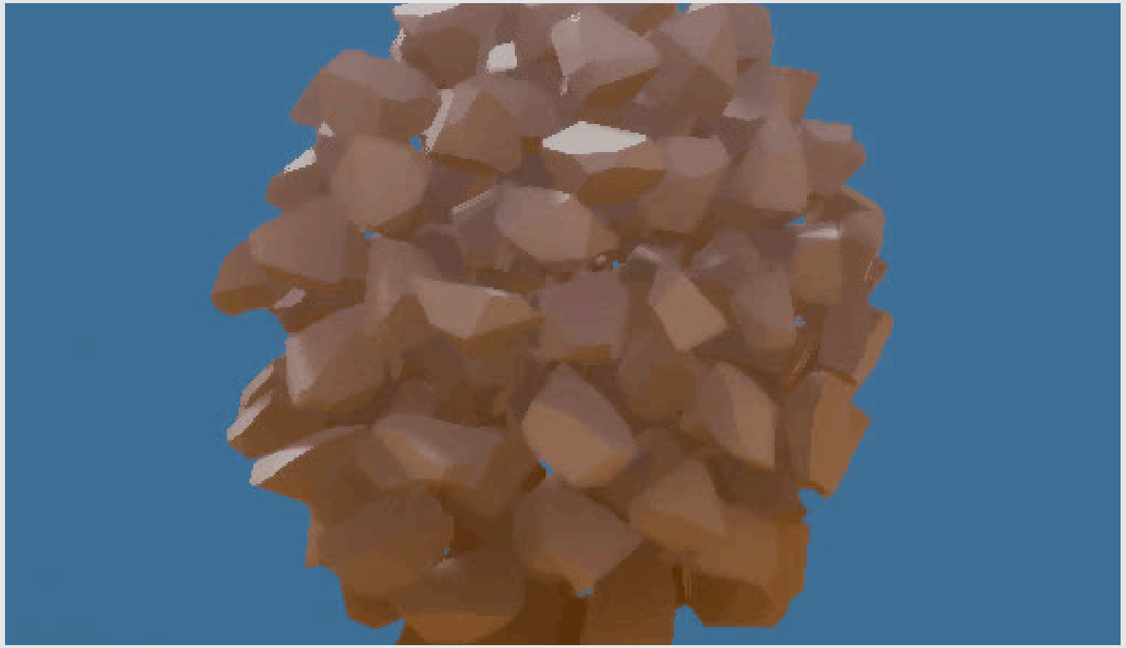
- Fonction qui est appelé pour chaque pixel d’un objet

Vertex shader :

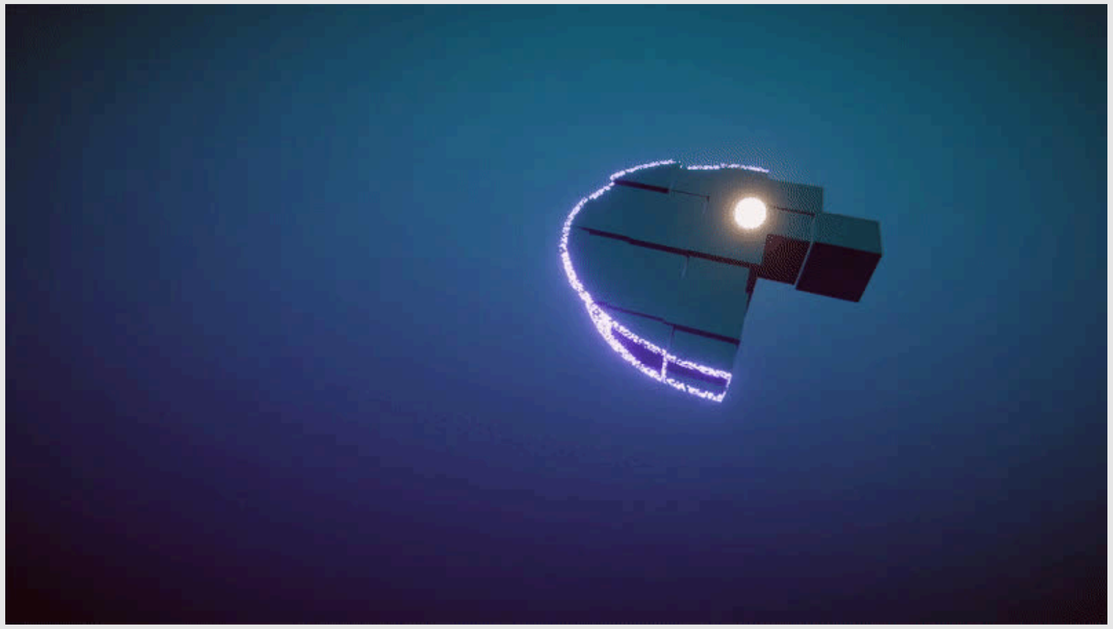
- Fonction qui est appelé pour chaque vertex d’un objet

Compute shader :

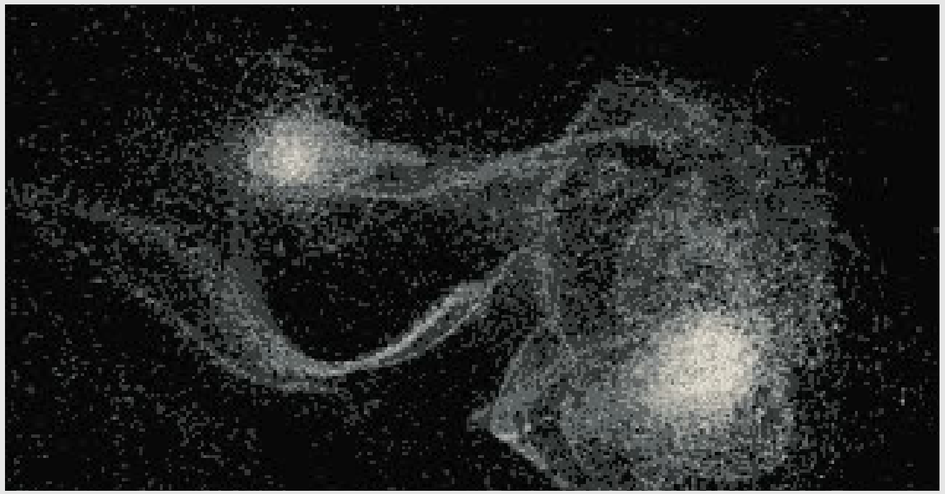
- Shader plus général pour faire des calculs parallèles



Exemple de vertex shader (l’unité de base c’est le vertex)



Exemple de fragment shader (l’unité de base c’est le pixel)



Exemple de compute shader (On travaille sur des étoiles pas sur des pixels ou des vertex)

04

Fragment shader

Fonction qui est appelé pour chaque pixel d’un objet

Utile aussi bien en 2D qu’en 3D :

- Y’a des pixels en 2D et en 3D

Permet de faire des effets pixels par pixels

- Plus gourmand en ressources que les vertex shaders (il y a beaucoup de pixels)

05

Vertex shader

Fonction qui est appelé pour chaque vertex d’un objet

Plus utile en 3D qu’en 2D:

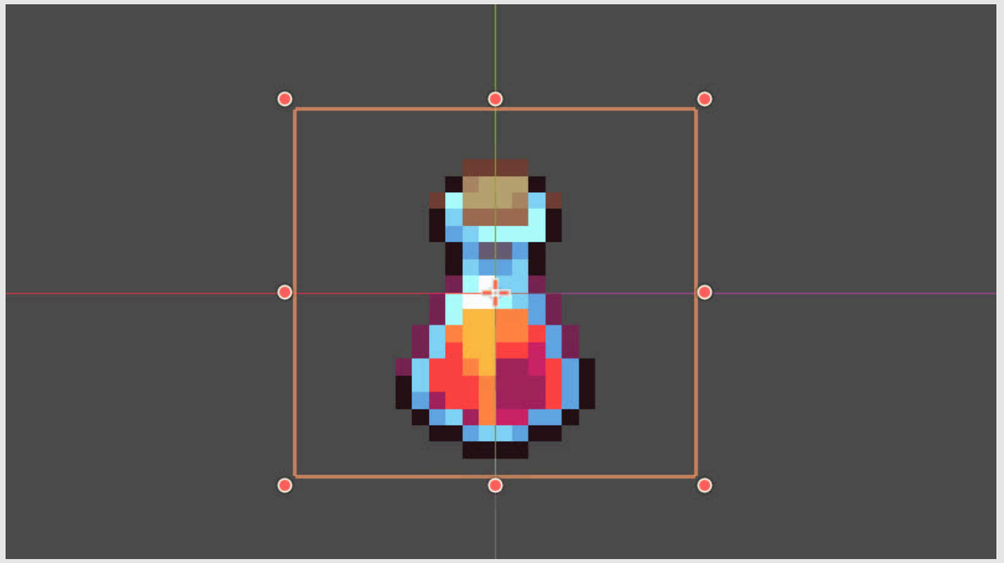
- Mais ça peut être utile en 2D ! Les objets en 2D ont aussi des vertex (pas beaucoup, 4 par exemples pour un rectangle)

Peut servir à faire des animations:

- Exemple du poisson

Plus performant que les fragments shaders :

- Y’a beaucoup moins de vertex que de pixels (ordre du millier vs ordre du million)



Présentation GLSL Godot

GLSL OpenGL Shading Language

Godot utilise un langage de programmation pour les shaders basé sur le GLSL avec du sucre syntaxique en plus

GLSL est un langage de programmation créer pour openGL en 2004 pour faire des shaders.

openGL c'est lent, mais pas de soucis Godot peut compiler son GLSL vers SPIR-V pour avoir des shaders rapide ensuite (avec Vulkan)

La façon dont les shaders est compilé puis interprété en Godot dépend de la méthode de rendering :

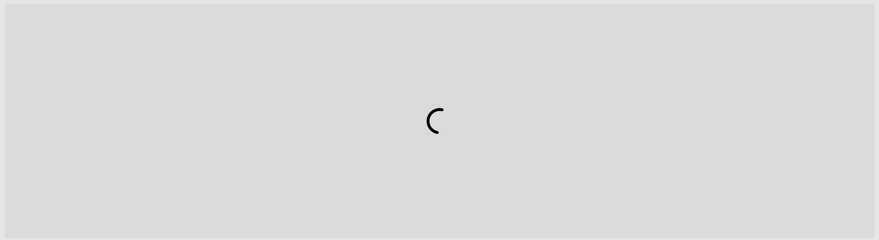
- Forward+ (Super performant sur nouvelles architectures, plein de features cools et modernes, utilise Vulkan)
- Mobile (Fait pour les appareils mobiles, pas mal de feautres cools, utilise Vulkan)
- Compatibility (Fait pour le web est les vieux PC (avant 2014) utilise OpenGL)

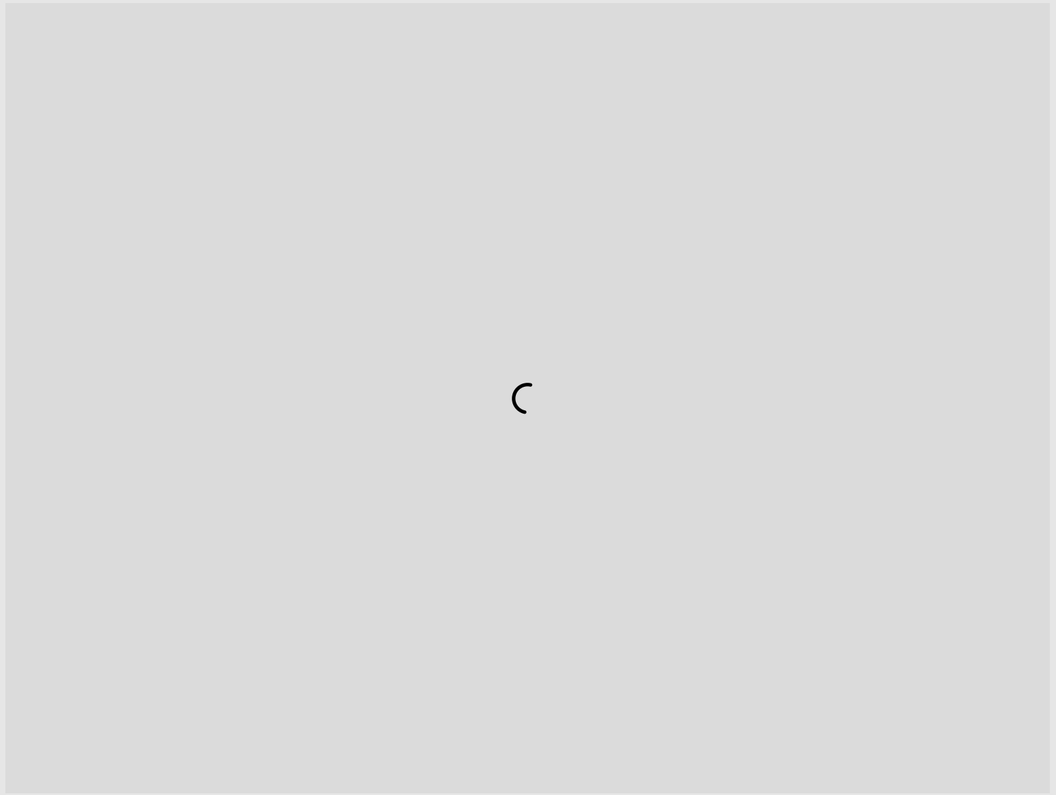
07

Exemple code shaders

Godot Shaders : Le site

- Plein de shaders stylés gratuitement avec leur code, partez de là pour faire vos shaders





08

Cheatsheet GLSL Godot

Variables globales utiles

Variable	Description
COLOR (in/out)	Couleur final du pixel (rgba)
UV (in/out)	Cordonnées UV du pixel (x,y) avec x et y entre 0 et I
FRAGCOORD (in)	UV mais par rapport à l'écran en entier
VERTEX (in/out)	Position du vertex dans le vertex shader
TIME (in)	Temps écoulé en seconds

Échantillonnage de texture :

```
vec4 tex_color = texture(TEXTURE, UV);
```

Uniform (équivalent de @export pour les shaders) :

```
uniform float omg_c_unfloat;
uniform sampler2D omg_c_une_texture;
```

Fonctions mathématiques et utilitaires

Fonction	Description
mix(a, b, t)	Interpolation linéaire entre a et b
smoothstep(a, b, t)	Interpolation smooth entre a et b
clamp(x, min, max)	Contraint x entre min et max
length(vec)	Longeur du vecteur
normalize(vec)	Renvoie vec normalizé
sin(x), cos(x)	Fonctions trigo
dot(a, b)	Produit scalaire entre a et b

08 Chad Cheetsheet GLSL Godot

https://www.reddit.com/r/godot/comments/16p8n2k/godot_shader_cheat_sheet_i_made_because_i_always/#lightbox