TER

29 mai 2024

# Simulation de fumée

Master 1 - IMAGINE

SERVA Benjamin - DELVIGNE Brian - HARCHA Ibrahim

# Table des matières

# Introduction

## Nos Motivations :

-Reproduire quelque chose de réel

-Découvrir un nouveaux monde de la 3D

## Les Objectifs :

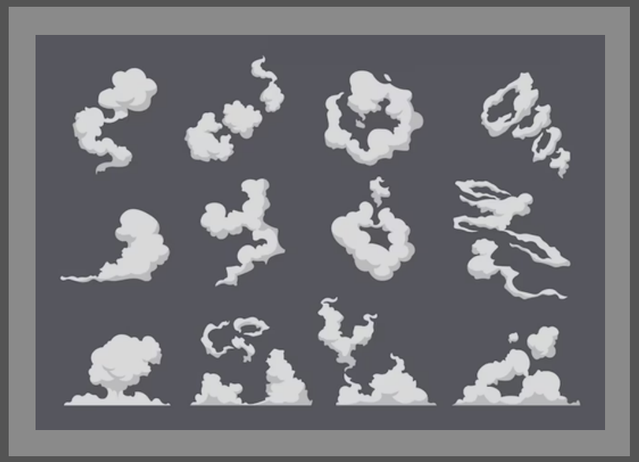
-Créer un générateur de particules

-Simuler de la fumée réaliste

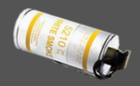
## Outils utilisés

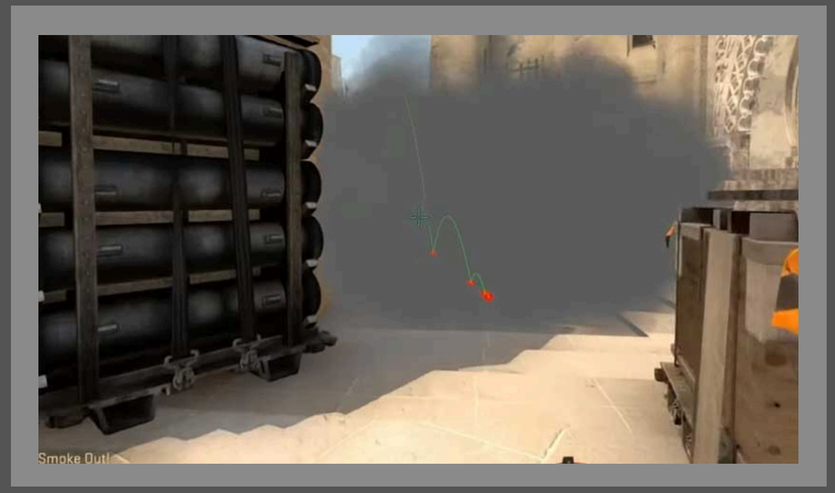

## Librairies utilisées

# Domaines d’utilisation

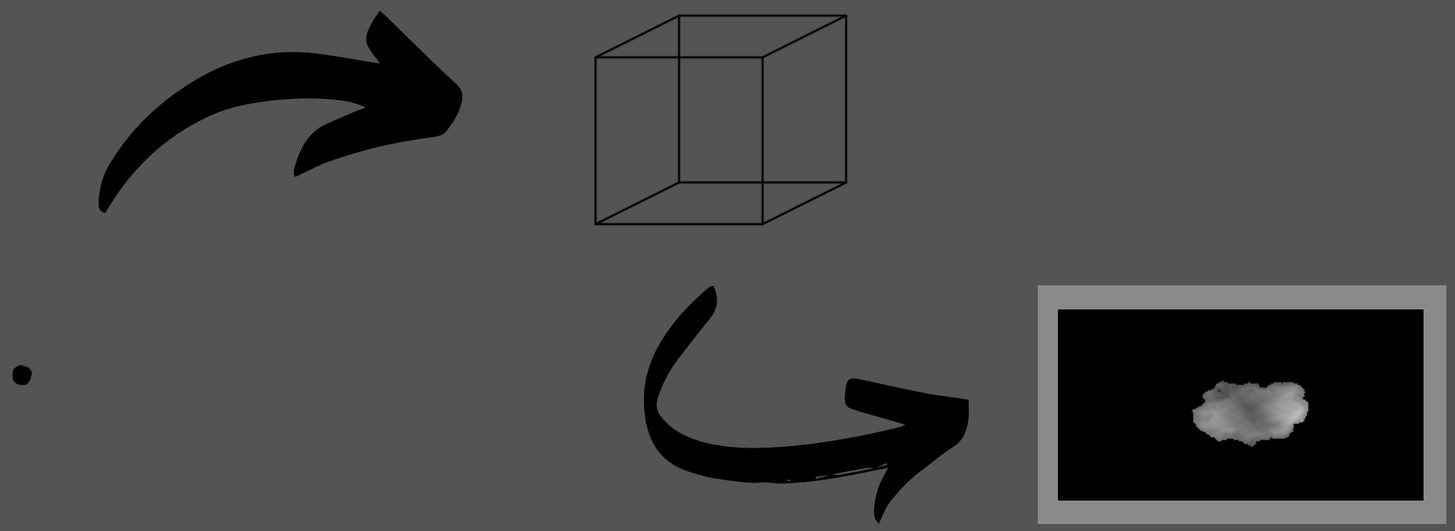


Différentes représentations de la fumée





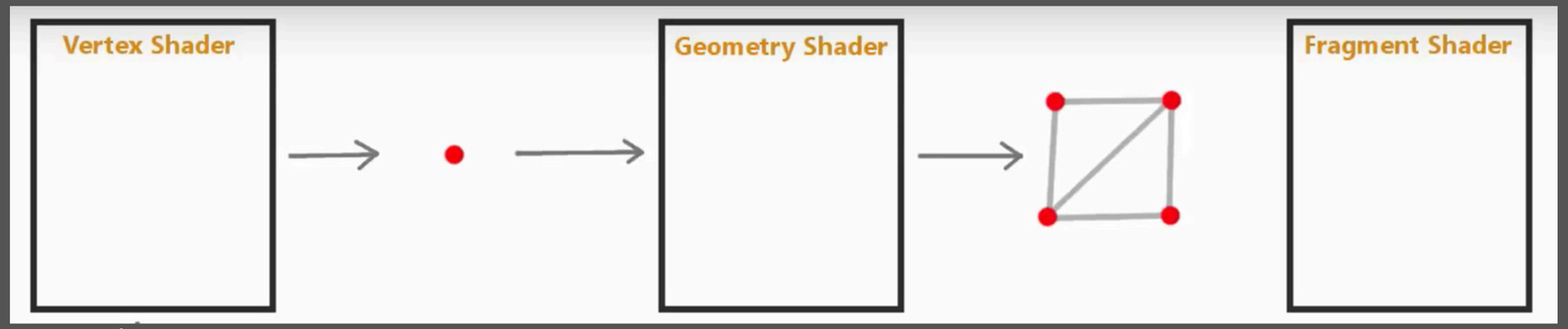
# Les Particules



```
struct Particle{
    std::vector<glm::vec3> position;
    std::vector<float> life;
    std::vector<glm::vec3> deplacement;
    std::vector<float> baseVelocity;
    std::vector<float> densite;
};
```

# Textures

## Utilisation du geometry shader

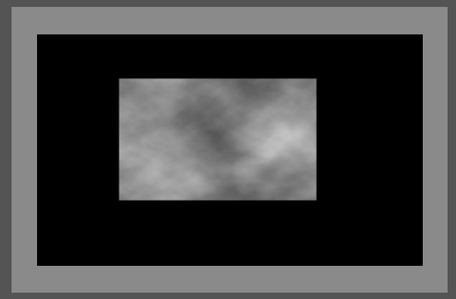


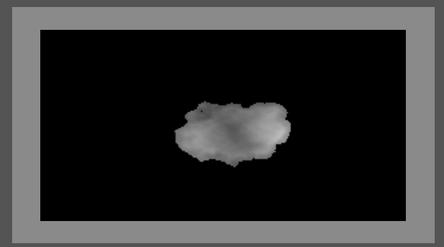

geometry_shader.glsl

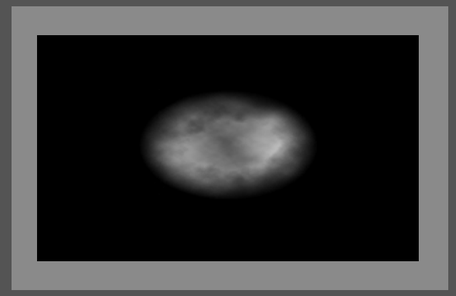
```glsl
#version 330 core
layout (points) in;
layout (triangle_strip, max_vertices = 6) out;

in float vertex_life[];
out float fragment_life;
out vec2 TexCoords;

uniform int size_p;

void main() {
    fragment_life=vertex_life[0];

    vec4 pos = gl_in[0].gl_Position;
    float size = 0.01*size_p; // size of the quad

    vec2 texCoordOffset = vec2(0.,0.);

    gl_Position = pos + vec4(-size, -size, 0.0, 0.0);
    TexCoords = texCoordOffset + vec2(0.0, 0.0);
    EmitVertex();

    gl_Position = pos + vec4(size, -size, 0.0, 0.0);
    TexCoords = texCoordOffset + vec2(1, 0.0);
    EmitVertex();

    gl_Position = pos + vec4(-size, size, 0.0, 0.0);
    TexCoords = texCoordOffset + vec2(0.0, 1);
    EmitVertex();

    gl_Position = pos + vec4(size, size, 0.0, 0.0);
    TexCoords = texCoordOffset + vec2(1, 1);
    EmitVertex();

    EndPrimitive();
};
```

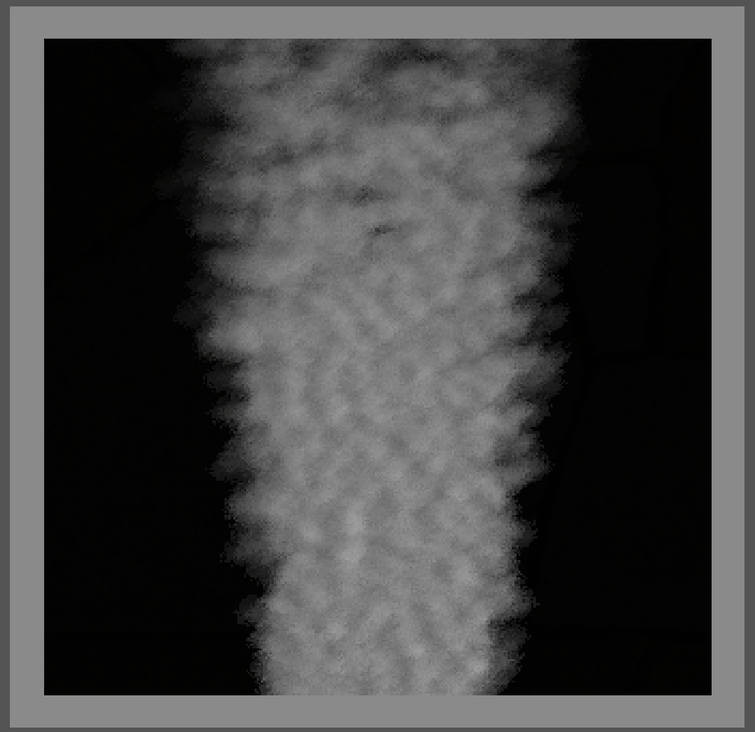
# Textures

## Gestion du canal alpha



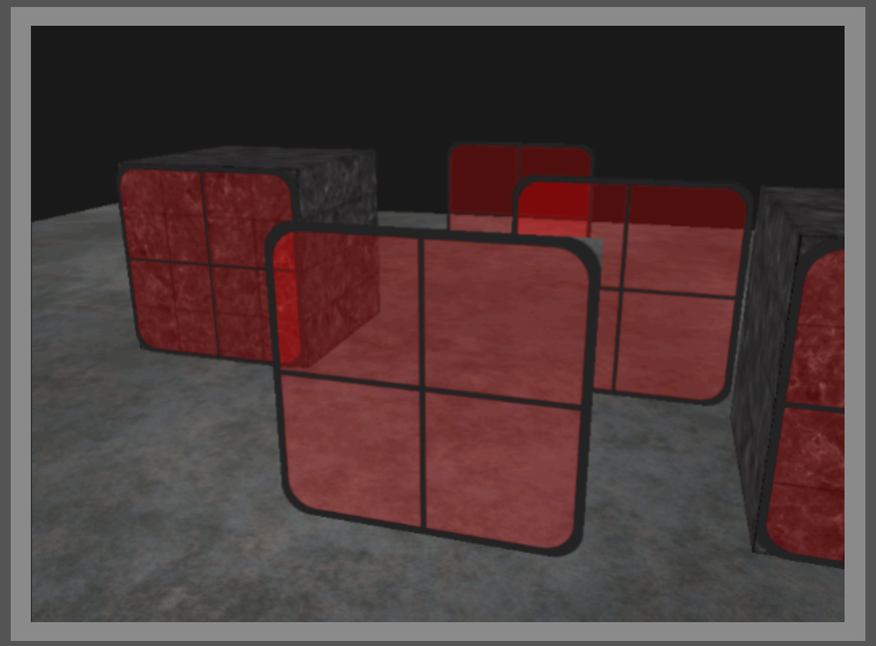


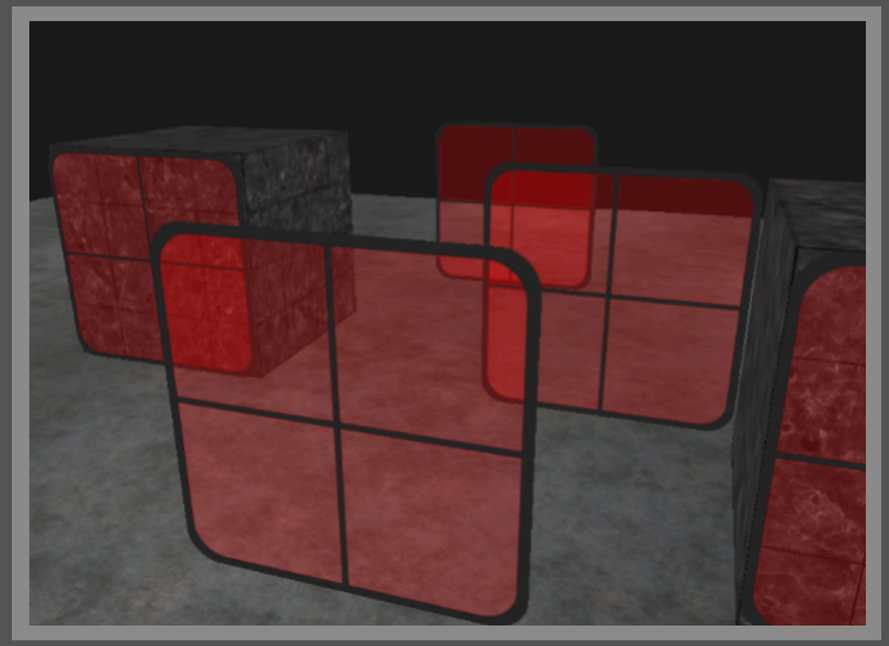




Gestion de la transparence basée sur la durée de vie

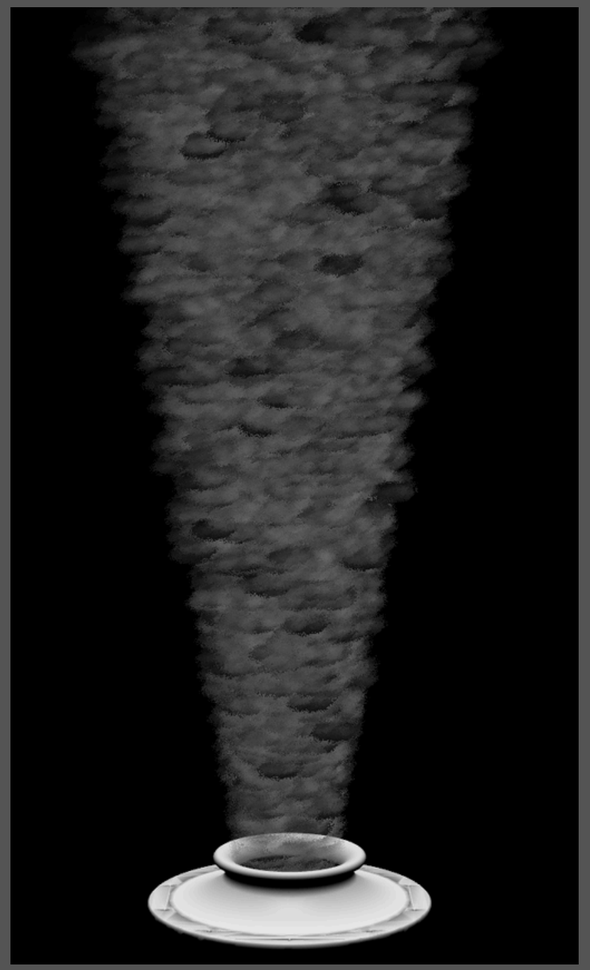
# Transparence



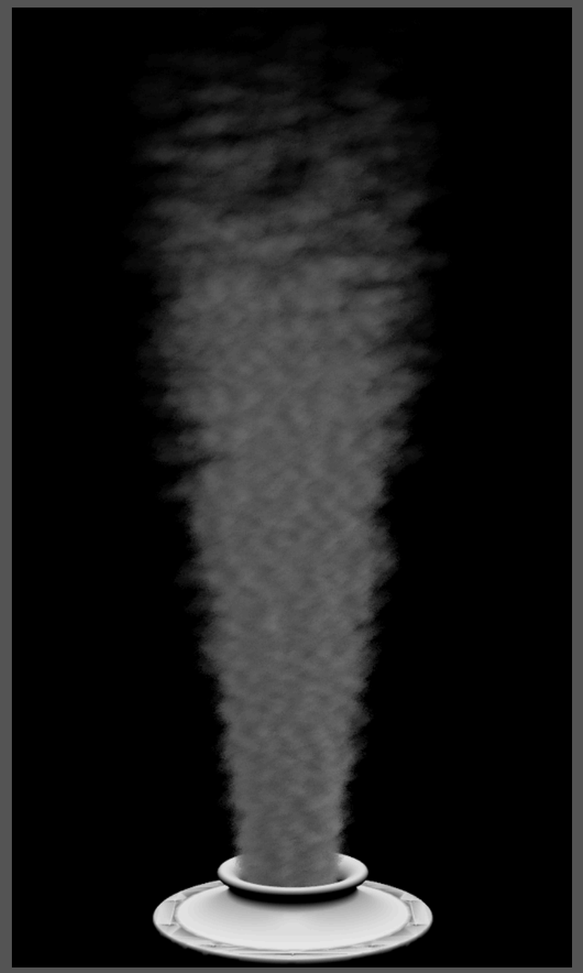


Exemple d’OpenGL

# Transparence



Dans le cas de notre fumée



# Bruit de Perlin

**Permet d’assurer une meilleure continuité entre les particules**

sans bruit

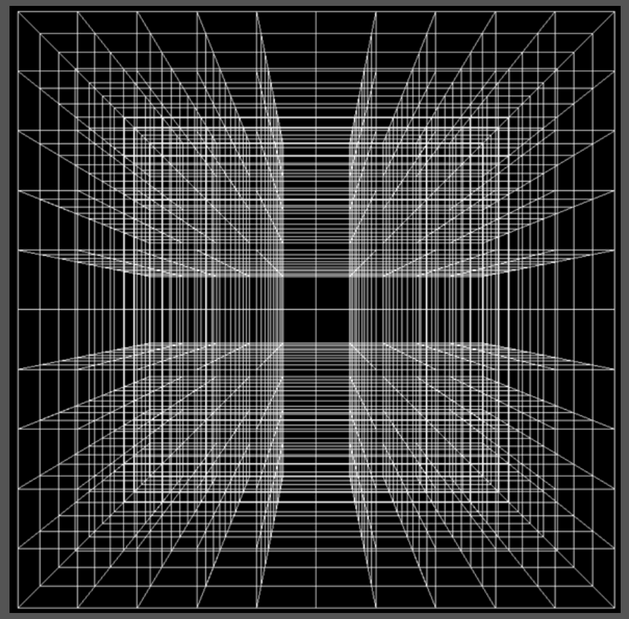
avec bruit d’amplitude 0.15

avec du bruit en excés

# Mouvement

Navier-Stokes

Gauss-Seidel

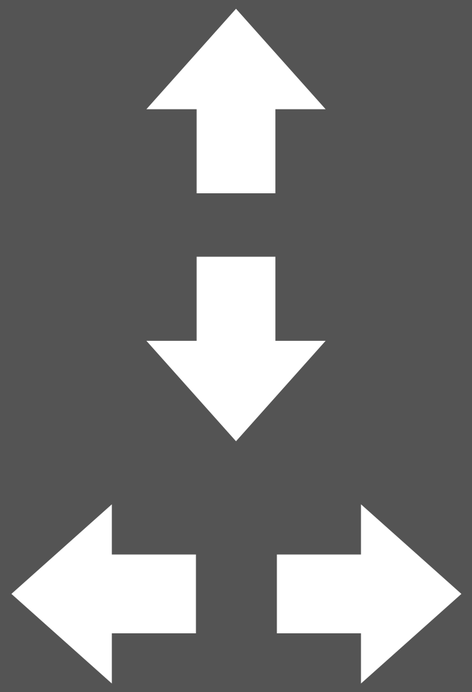


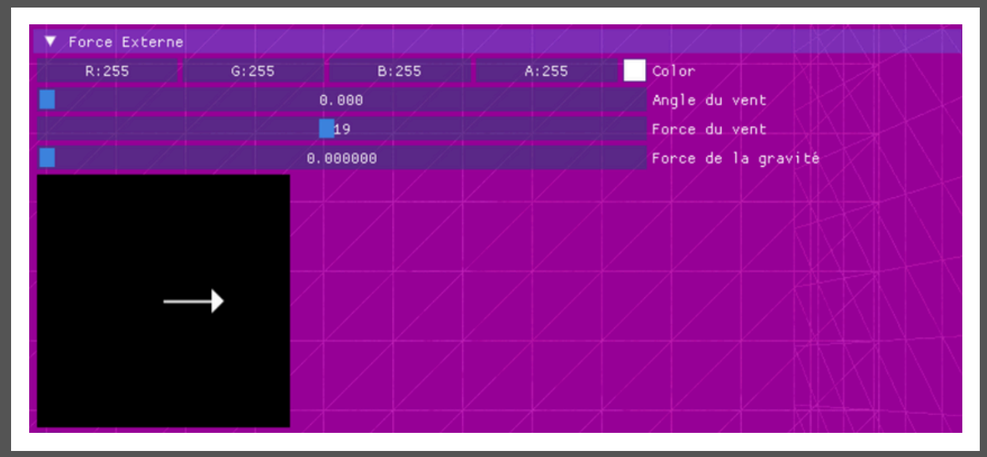
# Les forces externes

Expulsion

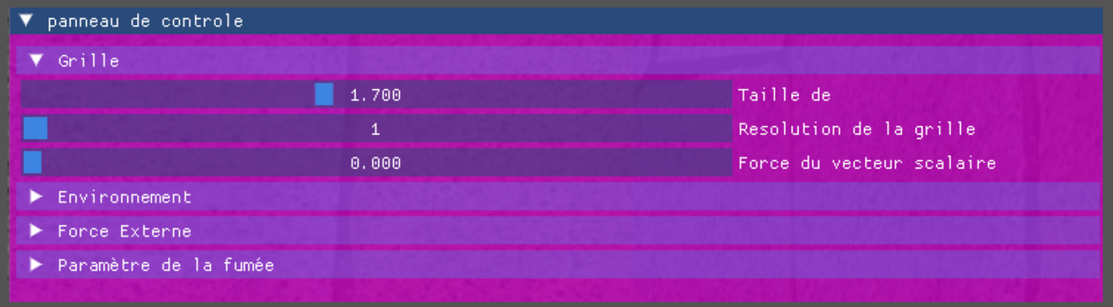
Gravité

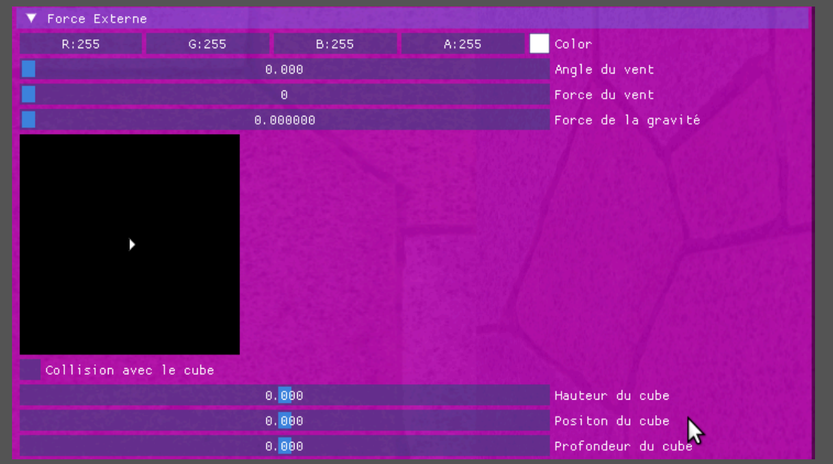
Perturbations

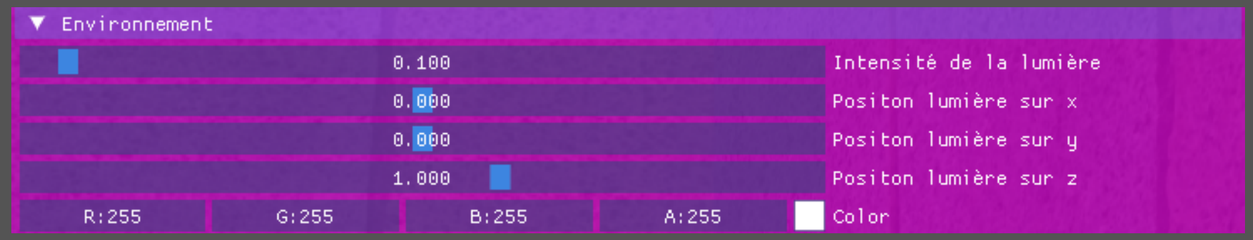




# Contrôle utilisateur




# Environnement

Illumination de Phong

normales

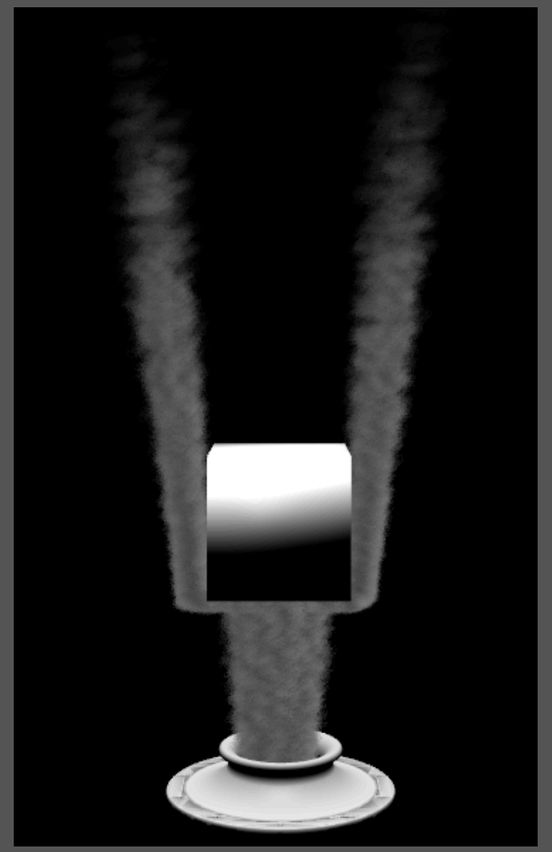
caméra

lumières

**Environnement externe à la fumée traité dans un shader différent**

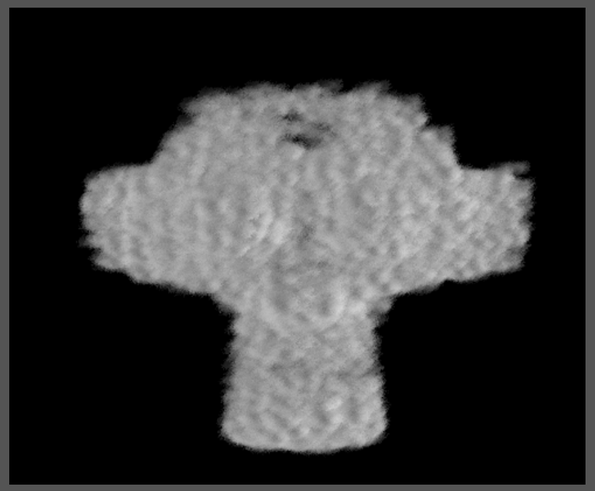
# Environnement

Interaction avec un objet externe : cube



Illumination de Phong

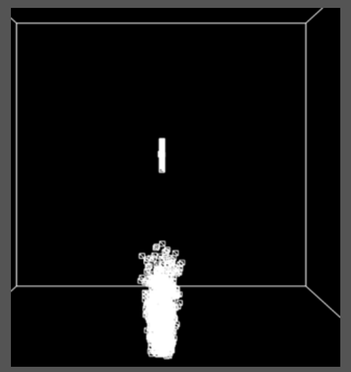
# Les Mesh



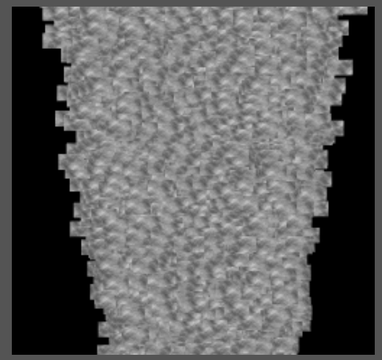
# Démonstration

# Problèmes rencontrés

```
struct Particle{
    glm::vec3 position;
    float life = lifeglobal;
    vec3 deplacement;
    float baseVelocity = 0.01f;
};
```



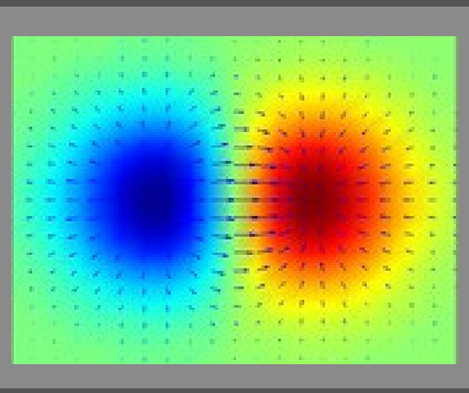
```
struct Particle{
    std::vector<glm::vec3> position;
    std::vector<float> life;
    std::vector<glm::vec3> deplacement;
    std::vector<float> baseVelocity;
    std::vector<float> densite;
};
```



**Canal transparence non pris en compte**

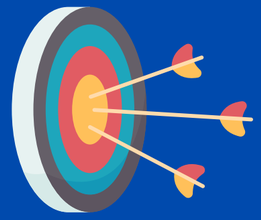
**Difficultés pour implémenter les algorithme de mécanique des fluides**

# Améliorations



# Conclusion

Avons-nous atteint notre objectif ?



Ce que nous avons appris

# Merci pour votre attention !

Avez-vous des questions ?

**Nous tenons encore une fois à remercier Mme Faraj pour son aide précieuse tout au long du projet**

# Références


. https://github.com/g-truc/glm - **GLM**

. https://www.glfw.org/ - **GLFW**

. https://github.com/ocornut/imgui - **DearImGui**

. https://learnopengl.com/ - **Système de particules et transparence**

. https://github.com/PacktPublishing/OpenGL-4-Shading-Language-Cookbook-Third-Edition