

# Initiation aux Shaders avec Processing

## Le hello World des shaders

### shader\_01

vert.glsl

```
uniform mat4 transformMatrix;

attribute vec4 position;
attribute vec4 color;

varying vec4 vertColor;

void main() {
    gl_Position = transformMatrix * position;

    vertColor = color;
}
```

frag.glsl

```
#ifdef GL_ES
precision mediump float;
precision mediump int;
#endif

varying vec4 vertColor;

void main() {
    gl_FragColor = vertColor;
}
```

shader\_01.pde

```
PShader myShader;
int margin = 32;

void setup() {
    size(800, 600, P2D);

    myShader = loadShader("frag.glsl", "vert.glsl");
    noStroke();
}

void draw() {
    background(0);
    shader(myShader);
    fill(230, 120, 0);
    rect(margin, margin, width - 2*margin, height - 2*margin);
    resetShader(); // Désactive le shader, permet de redessiner normalement
}
```

## Communication entre l'application et les shaders

L'application (programme Processing) peut envoyer des données vers les shaders par des variable déclarées avec le mot-clé 'uniform'. ===== Fonctions Processing pour transmettre des données =====  
 set(String name, int x) → vec2 set(String name, int x, int y) → vec3 set(String name, int x, int y, int z) → vec4 set(String name, float x, ...) set(String name, PVector vec) → vec3 set(String name, int[] vec, int ncoords) Jusqu'à 4 coordonnées par élément  
 set(String name, float[] vec, int ncoords) set(String name, PMatrix2D mat) → mat2 set(String name, PMatrix3D mat) → mat4 set(String name, PImage tex) → sampler2D Attention ! Lorsqu'on transmet des nombres entiers, comme par exemple : set("u\_resolution", 512, 512), soyez sûr d'avoir déclarée les variables uniform pour des types entiers, comme : ivec2, 'ivec3... Une autre solution est de les convertir en nombres flottants avant de les transmettre : set("u\_resolution", float(512), float(512))' ===== Affichage de textures =====

Article extrait de : <http://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : <http://lesporteslogiques.net/wiki/ressource/code/processing/shaders?rev=1662473559>

Article mis à jour: **2022/09/06 16:12**