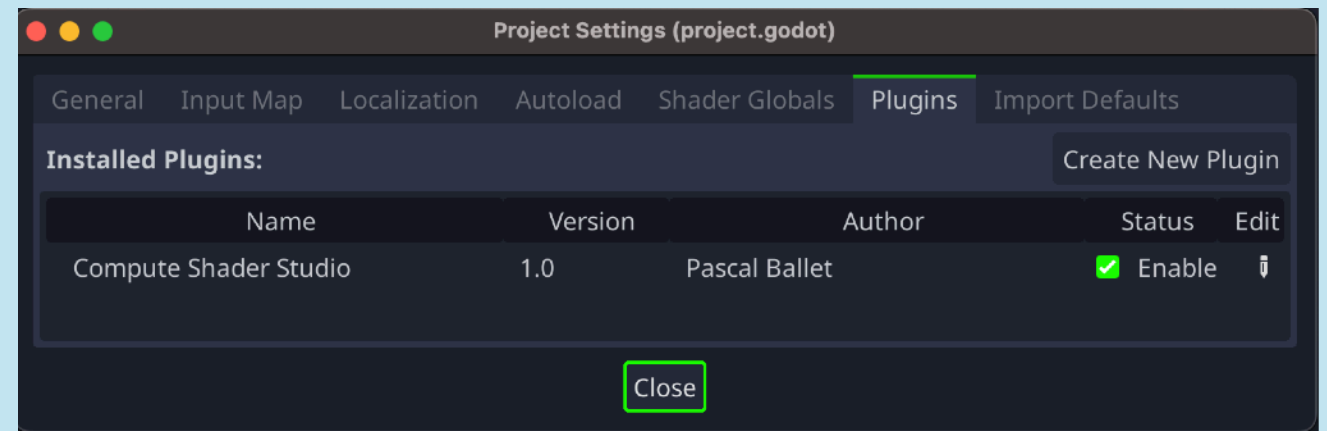Pascal BALLET

COMPUTE SHADER STUDIO
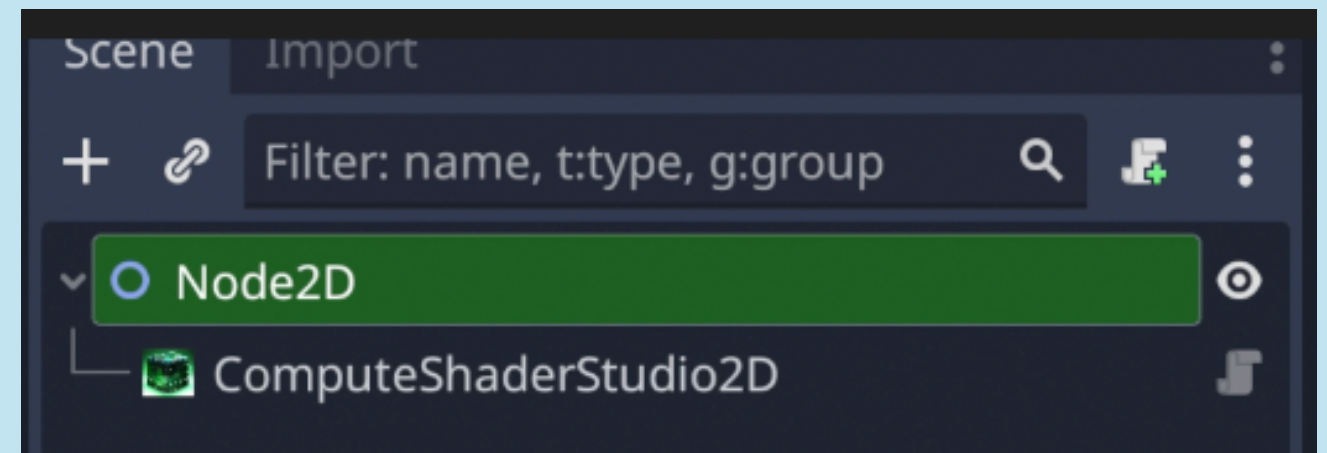
A Godot Game Engine Asset

Compute Shader Studio

TUTORIAL

- A 2 minutes video of this tutorial can be found here

1. Download and install **Compute Shader Studio** from the **AssetLib** tab of **Godot Engine**.

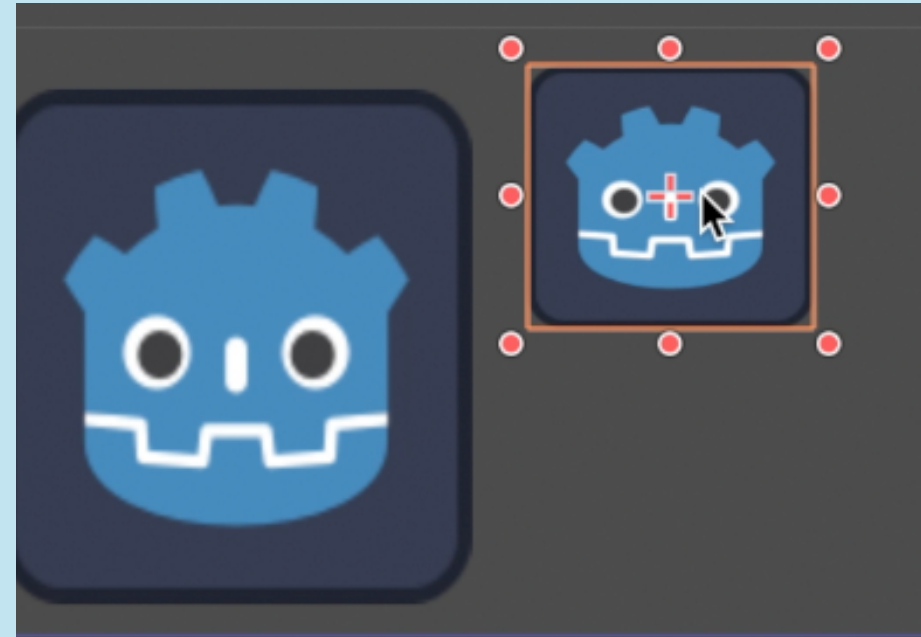2. Click on **Extensions** then click on « **Enable** »

3. Create a new **scene 2D**

4. Click the **+** button to add a new **ComputeShaderStudio node**.
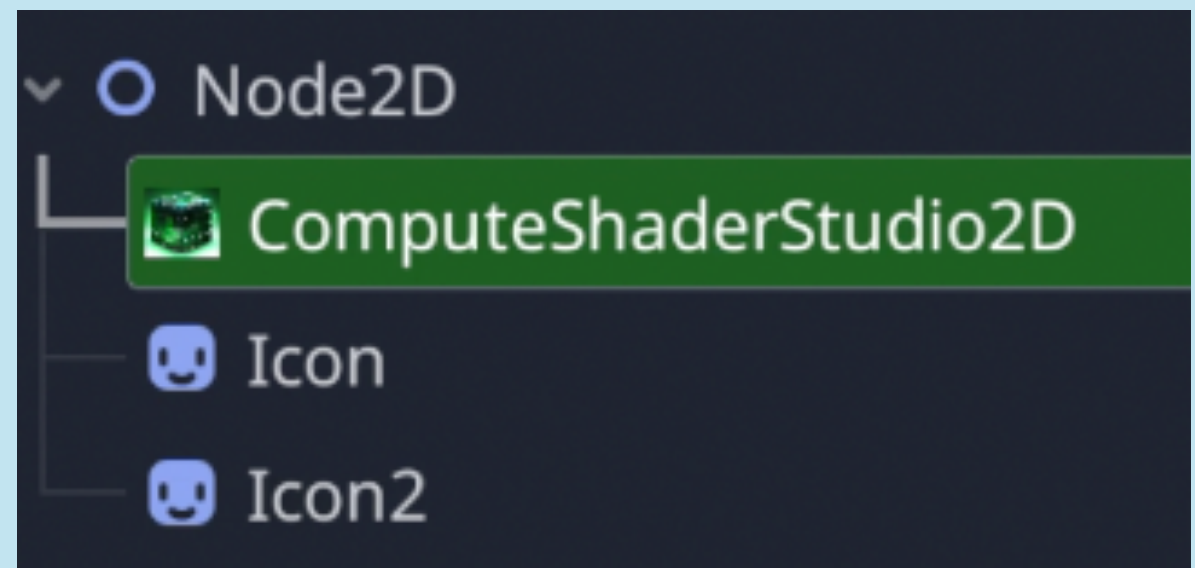
Compute Shader Studio

TUTORIAL

- Add the display with

  - Two icons (or Sprite2D)

  - Note you can add one, two, three or mode displays, according to your shader.

5. From the FileSystem, Drag & Drop **two icon.svg** in your 2D scene.

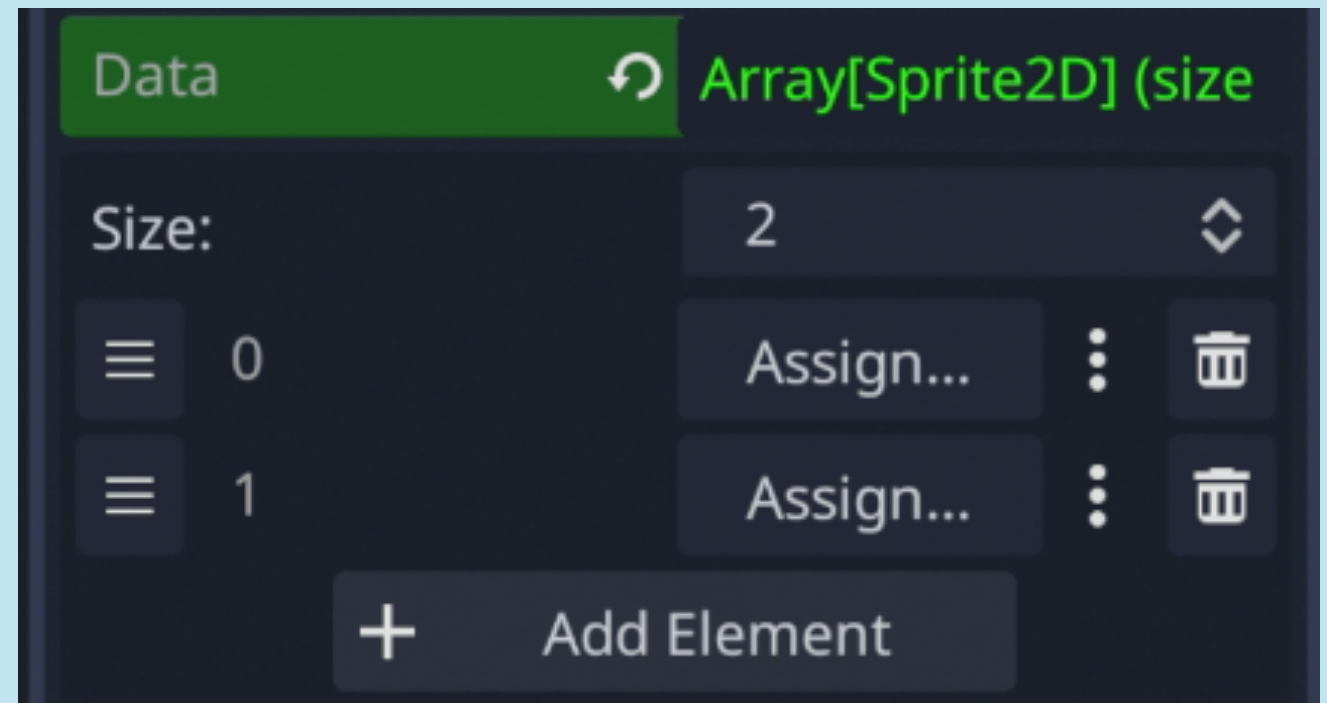6. Select the ComputeShaderStudio node

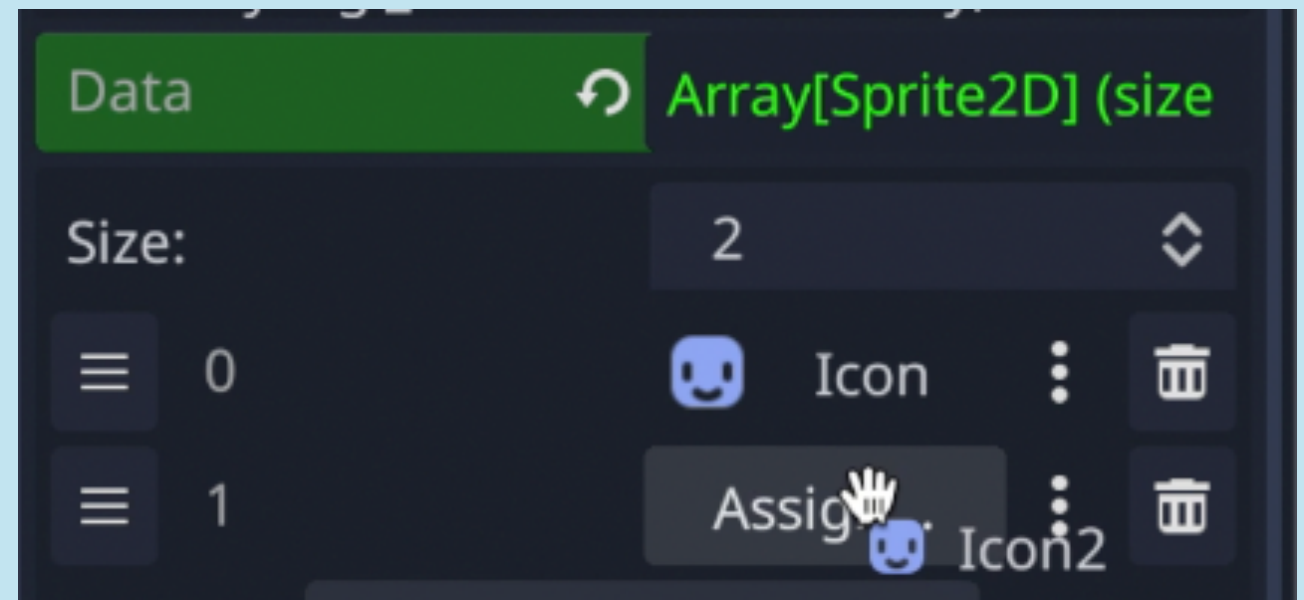# Compute Shader Studio

TUTORIAL

- The number or Data MUST be the same as the number of Sprite2D (will be changed in the future)

7. Then add **two Data** from its Inspector

8. **Drag & Drop the two icons** from the Scene Tree to the Inspector then **Execute** your scene.

Compute Shader Studio

TUTORIAL

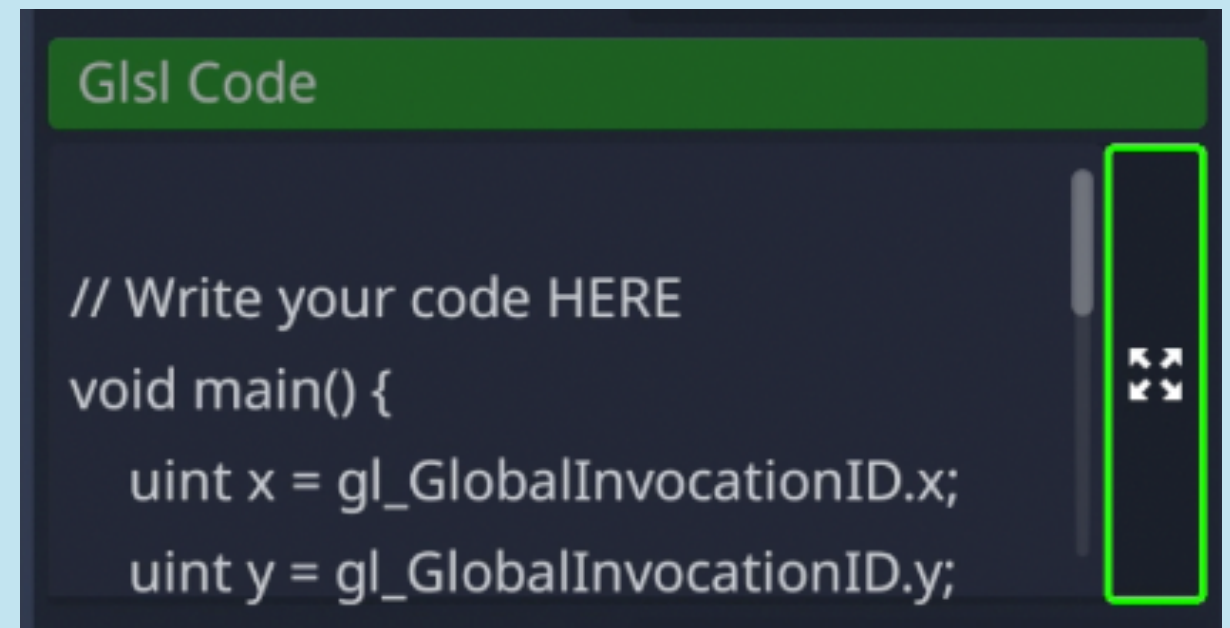- **Open the multiline editor for your GLSL code**

- **Or write it inside a more complete code editor then copy-past it in the Inspector « Glsl Code » property**

9. Now you can **run your scene**
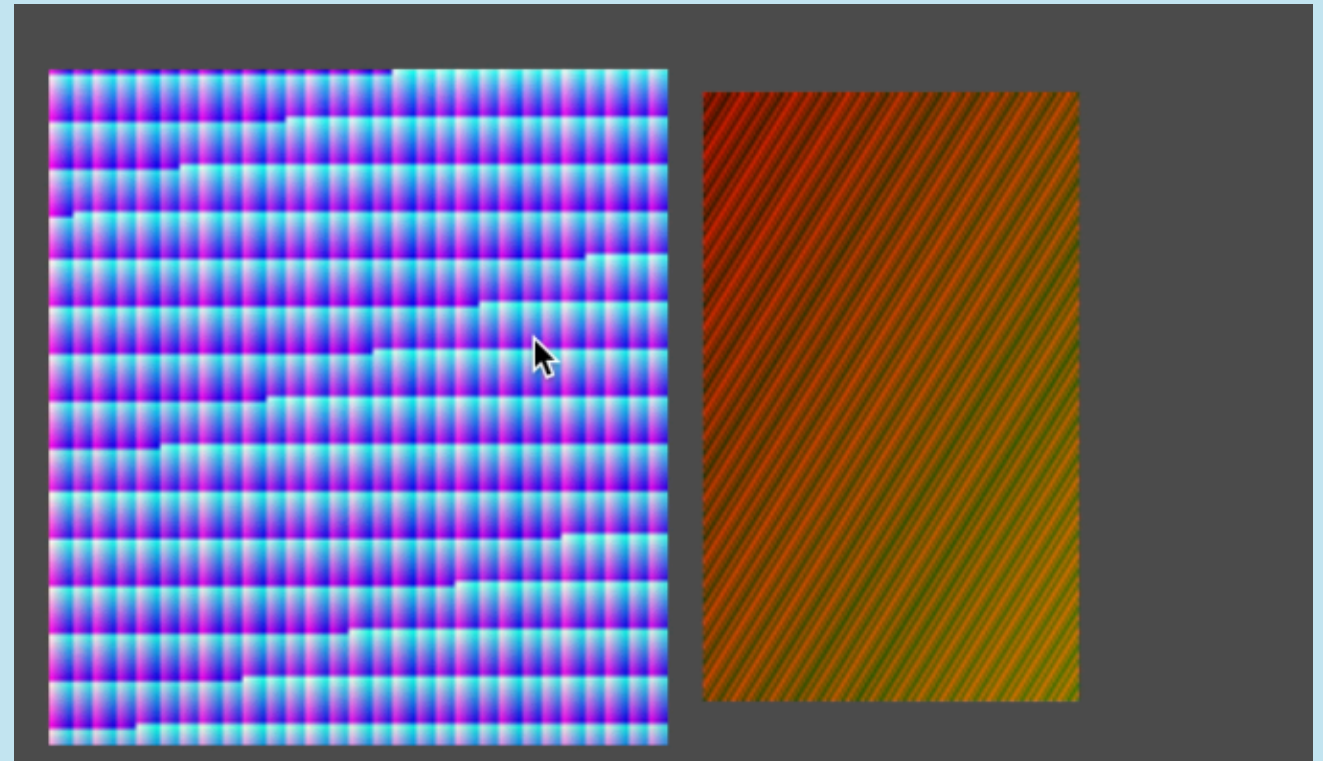
10. And change the **shader code** as you wish from the **Inspector**


```
Glsl Code

// Write your code HERE
void main() {
    uint x = gl_GlobalInvocationID.x;
    uint y = gl_GlobalInvocationID.y;
```

Compute Shader Studio

TUTORIAL

- **The Pause property allows you to control when you want the shader to compute**

- **Two methods can be connected to buttons to Play or make a simulation Step (see examples 3 and 4 in the Addon)**

11. You can change some parameters in the Inspector
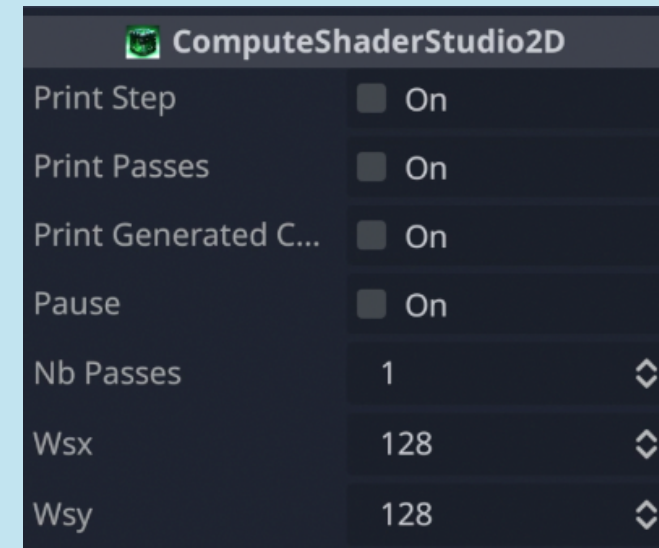
Nb Passes indicates how many **successives treatments** your program requires. Each pass wait for the whole execution of each process (Rendez-vous) before it executes the next pass. It is like a **barrier** between each pass.

Wsh and **Wsy** are the **Workspace** X and Y size of the execution processes. Generally it MUST be the SAME as the **Data sizes:** each process works on a single Data (pixel).

Print Step show the execution step in the **Output**

Print Passes show the execution pass in the **Output**

Print Generated Code display in **Output** the **entire GLSL code** made by **ComputeShaderStudio** to run your shader.

Compute Shader Studio

TUTORIAL

- **Make great compute shaders and unleash all the power of your computer :-)**

12. Look at the different examples available in the Addon to see what can be done with Compute Shader Studio

Compute Shader Studio - Example 2

```
// Write your GLSL code HERE
void main() {
    uint x = gl_GlobalInvocationID.x;
    uint y = gl_GlobalInvocationID.y;
    uint p = x + y * WSX;
    int opacity = 0xFF000000 ;
    int blue = 0x550000;
    int green = 0x3300;
    int red = int(y*2) ;
    data_0[p] = opacity+blue+green+red;
}
```

Compute Shader Studio
Example 4: BIG Game Of Life

Step

Play