

# Seamless Shadergraph Extension

Shadergraph is a powerful node-based shader editor. Since unity 2021.2, shadergraph is available on any pipeline. This extension offers multiple additional nodes such as new nodes including a baking node to use shadergraph as a procedural texture baker.

## Why this extension exists ?

---

Writing a custom shader for many object in a scene is not always great ; sometimes custom shaders can fail building on specific platforms, be unnecessarily slow and are not always cross pipeline compatible. This extension can transform shadergraph into a texture editor. The output of a texture editor is one or more **textures**.

It means that you can **use these textures with Unity default materials and get profit of the cross platform guarantee** and great performances on **any pipeline**.

As the generated textures are PNG, you can use them in custom shaders too, or even non unity-related purposes such as making a logo.

## Getting started

---

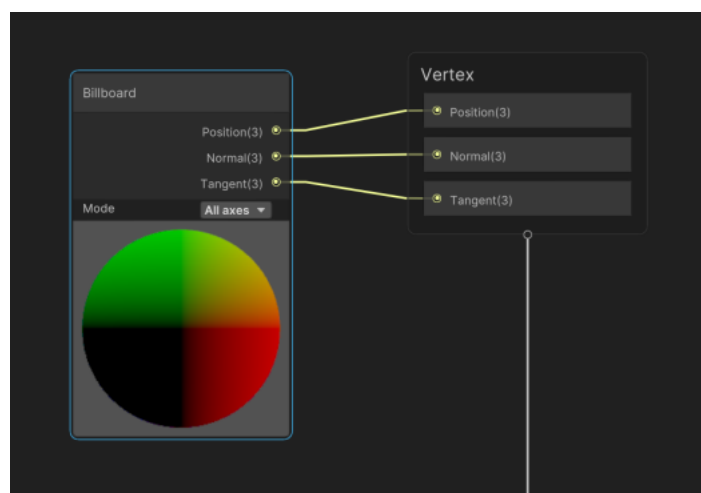
Once you purchased the asset on the asset store, you can import it using the package manager. Make sure to import shadergraph too if it is not the case already.

This extension works on any pipeline, but the demo materials need to be updated to URP or HDRP if you don't use the built-in render pipeline.

After importing the package, you should already see a new category called "Seamless" in the node list in shadergraph.

## Billboard effect

---



The provided Billboard node

Seamless SGE embeds a billboard node to apply [billboarding](#) on any object by displacing its vertices in the vertex shader.

Once applied, the object will face the camera, either on all axes or just around the vertical axis according the "Mode" parameter.

As this takes place in the shader, it will work with several cameras too.

## Primitives

---

In order to bake tileable textures, it is necessary to use tileable primitive nodes. All the following nodes are the same than the ones in the original [Seamless](#) asset, please refer to it for more details.

### Voronoise

This noise [created by Inigo Quilez \(https://www.shadertoy.com/view/Xd23Dh\)](https://www.shadertoy.com/view/Xd23Dh) combines standard and voronoi noises in the same place. You can control how much it look look squared or voronoi-like, as well as sharp/blurry.

### Voronoi Ultimate

A more advanced Voronoi noise than the one provided by unity.

### Fractal Noise

Same as Voronoise, but the fractal version. you can apply up to 16 octaves with control over the gain and lacunarity.

Fractal noises can be really resource consuming, better use it for baking purpose only.

### Fractal Warp

Same as fractal noise with [domain warping \(https://iquilezles.org/articles/warp/\)](https://iquilezles.org/articles/warp/) applied.

### Faded Polygon

A simple polygon with an inner falloff control to blur the edges.

### Faded Rounded Rectangle

A simple rounded rectangle with control over the radius, and an outer falloff.

## Squircle

A [Squircle](#) with control over the convexity and an inner falloff.

## Caustics Noise

A Fractal noise that looks like underwater caustics

## Erosion Noise

A Fractal noise that looks like mountains erosion

## Filter nodes

---

### Curvature From Height

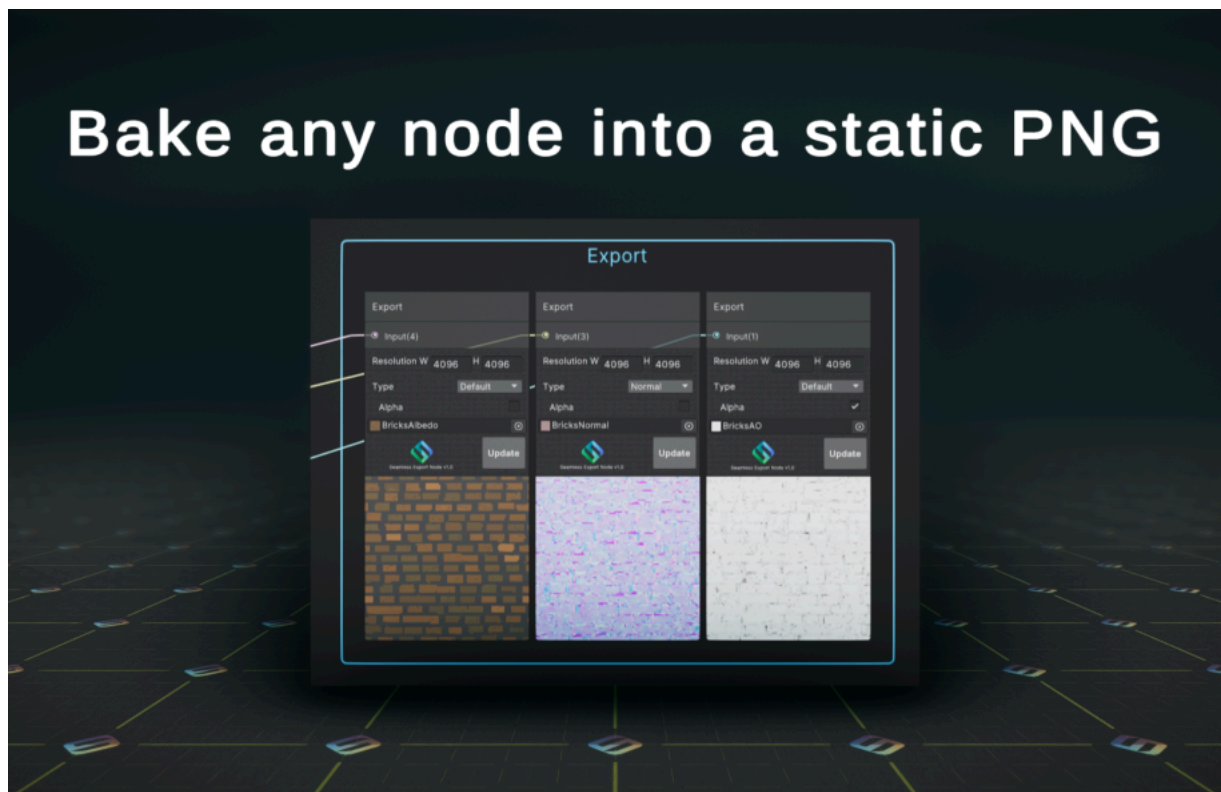
The curvature filter is a must-have to get a nice ambient occlusion estimation.

## Luminance

A simple RGB to Luminance helper node.

## Export node

---



The export/baking node

The export node is another useful node this extension provides. You can plug any output and bake the texture from the shader generated at this node. You can chose any size you want, including non-square or non-POT textures.

You can chose if you want to include the alpha channel, and what type of texture it should be imported as (Default, Normal Map or Raw).

Only PNG and RAW exports are supported yet.

RAW format is natively compatible with terrain height-map import.

## Support

---

I will be glad to help if you encounter any issue.

Discord (<https://discord.gg/kYwzdVAt8q>)

Email (<mailto:support@leochaumartin.com>)

---

---