

The intention of this document is to describe how to use the Fast Noise Generator (FNG) inside Unreal Engine and all of its settings. All the blueprints reference will be included here too.

The FNG has been created by Víctor Hernández Molpeceres using the [Auburn's Fast Noise library](#). It supports UE4.22 and beyond.

Fast Noise library

Fast Noise is an open source noise generation library with a large collection of different noise algorithms. It's used, for example, to calculate the noise that will be applied on the terrain height in the [Procedural Terrain Generator](#) plugin.

Usage

The first thing you will need is a **Fast Noise Wrapper** reference, then you can call the **Setup Fast Noise** function to configure these settings:

- Noise Type - Algorithm used to generate the terrain height.
- Seed - Seed used for all noise types. Using different seeds will cause the noise output to change.
- Frequency - Frequency for all noise types, except White Noise. Affects how coarse the noise output is.
- Interpolation - Interpolation method used to smooth between noise values in Value and Perlin Noise. Possible interpolation methods (lowest to highest quality): Linear; Hermite; Quintic.
- Fractal Type - Method for combining octaves in all fractal noise types.
- Fractal Octaves - Octave count for all fractal noise types. The amount of noise layers used to create the fractal.
- Fractal Lacunarity - Octave lacunarity for all fractal noise types. The frequency multiplier between each octave.
- Fractal Gain - Octave gain for all fractal noise types. The relative strength of noise from each layer when compared to the last.

- Cellular Jitter - Maximum distance a cellular point can move from its grid position. Setting this high will make artifacts more common.
- Cellular Distance Function - Distance function used in cellular noise calculations. The distance function used to calculate the cell for a given point. Natural is a blend of Euclidean and Manhattan to give curved cell boundaries.
- Cellular Return Type - Return type from cellular noise calculations.

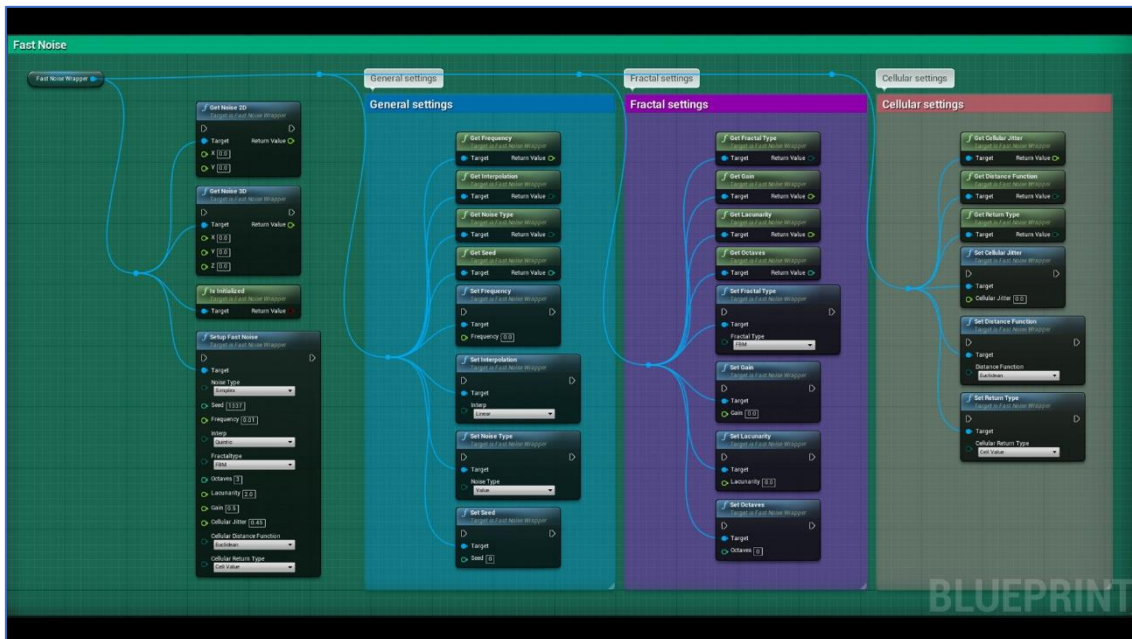
Now that you have it configured, you can call the **Get Noise 2D** and **Get Noise 3D** functions to obtain noise values, which can be used to generate height maps, for example.

C++

All the code is well-documented and can be used from other source code files in your projects. You can find and analyze it in `FastNoiseGenerator\Source\FastNoiseGenerator`

Blueprint reference

The FNG is also blueprint-friendly! All the settings and functions are also exposed to blueprints. You can find the functions under “Fast Noise” category.



Acknowledgements

Thanks to Auburns for creating the incredible Fast Noise library.