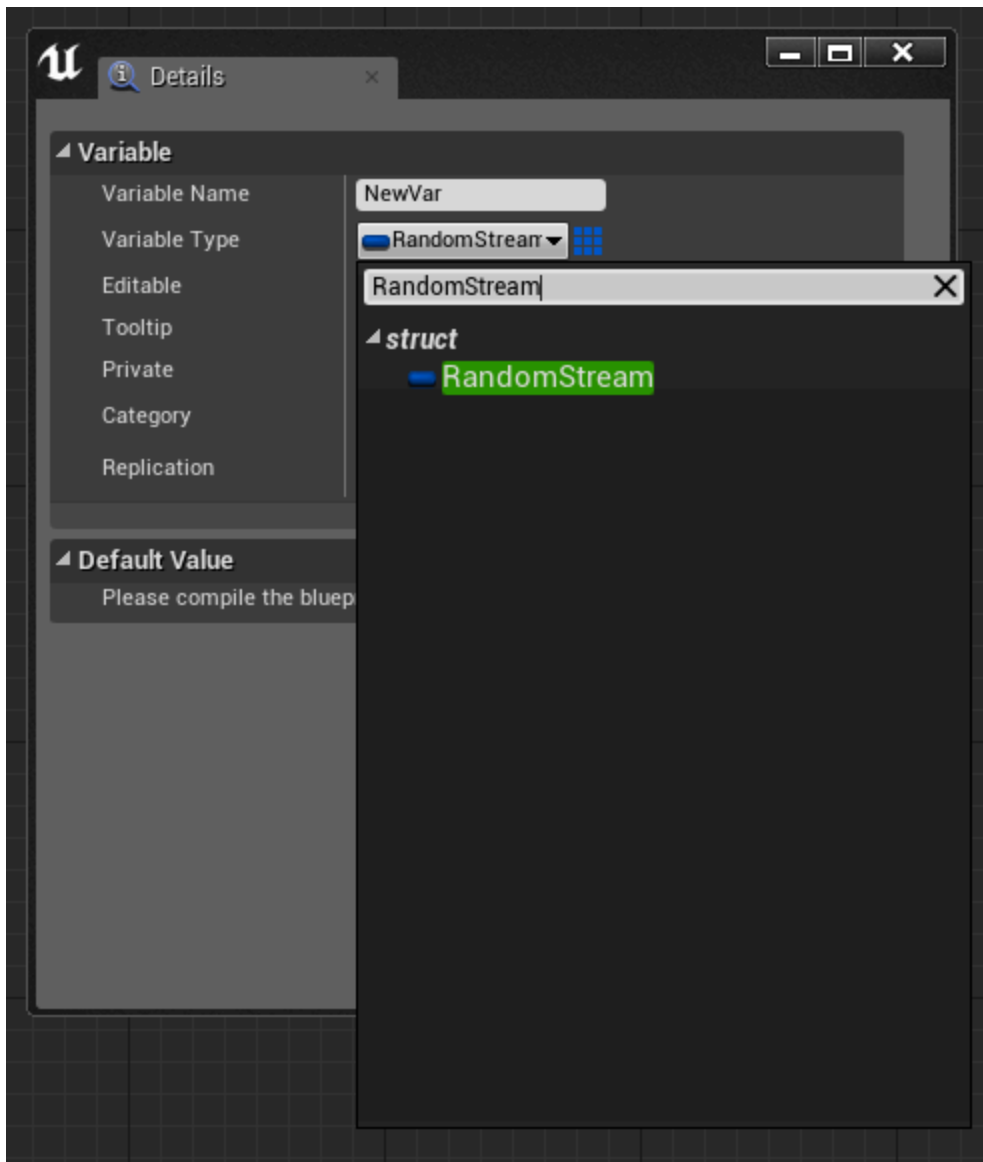# Random Streams

Explanation of random streams and how to use them in Blueprints.

RandomStreams allow repeatable, random numbers to be generated and used inside of Blueprints and Level Blueprints, as well as AnimBlueprints for animations. This can be extremely useful when setting up effects, such as scattering objects or building procedural environments, where you want a random effect, but also a consistent distribution each time the graph is evaluated. Previously, using random values would result in a different distribution each time the graph was evaluated, meaning moving the Blueprint or performing any action which caused the graph to be re-evaluated resulting in a completely different effect. With RandomStreams, you can tweak the effect based on a seed value to get the desired result, and then perform any other modifications you want while maintaining the overall effect.

RandomStream Variables

RandomStreams are available as a special type of struct variable in Blueprints. They can be added like any other variable by creating one through the **My Blueprint** tab in **Graph** mode.

See [Creating Variables](#) for complete instructions on how to add new variables inside Blueprint Classes or Level Blueprints.
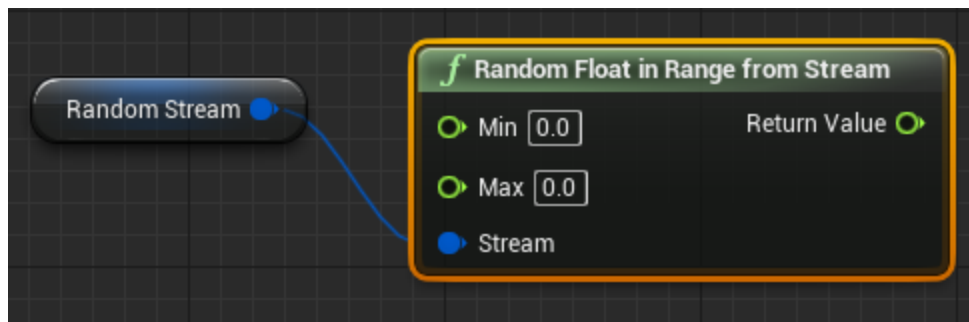
RandomStream Functions

In order to make use of a RandomStream variable, there are a set of functions which take a RandomStream as input and output a random value of different types, depending on the function.
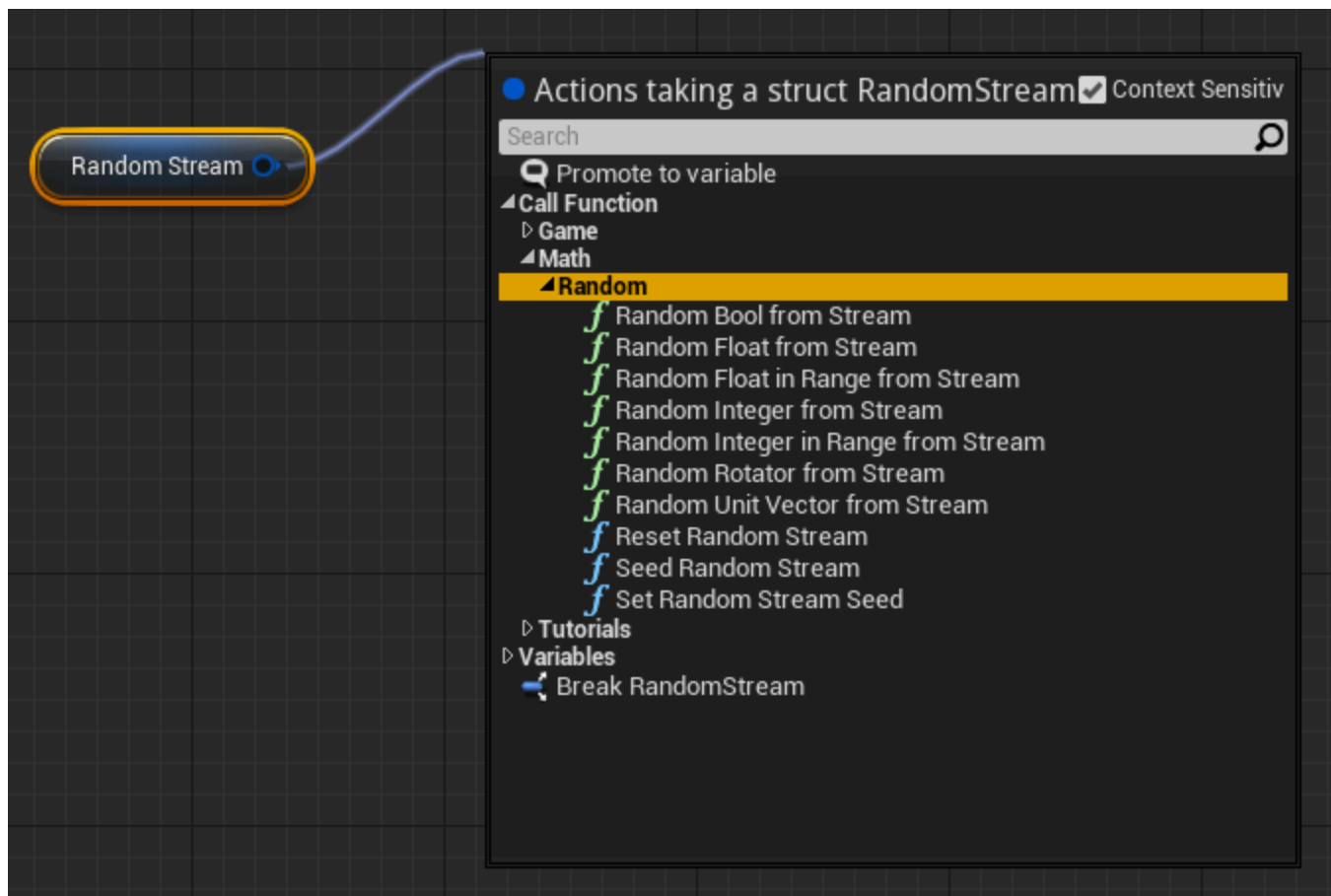
Function	Description
Random Bool from Stream	Outputs either 0 or 1 at random.
Random Float from Stream	Outputs a random Float value in the range (0.0, 1.0).
Random Float in Range from Stream	Outputs a random Float value in the range (Min, Max).
Random Integer From Stream	Outputs a uniformly distributed random Integer value in the range (0, Max - 1).
Random Integer In Range From Stream	Outputs a random Integer value in the range (Min, Max).
Random Rotator From Stream	Outputs a random Rotator value.
Random Unit Vector From Stream	Outputs a random unit-length Vector value.

Choosing one of the functions shown above will place it in the graph wired to the RandomStream variable.

An example of using RandomStream functions to randomly place and rotate a component belonging to the Blueprint is shown below:
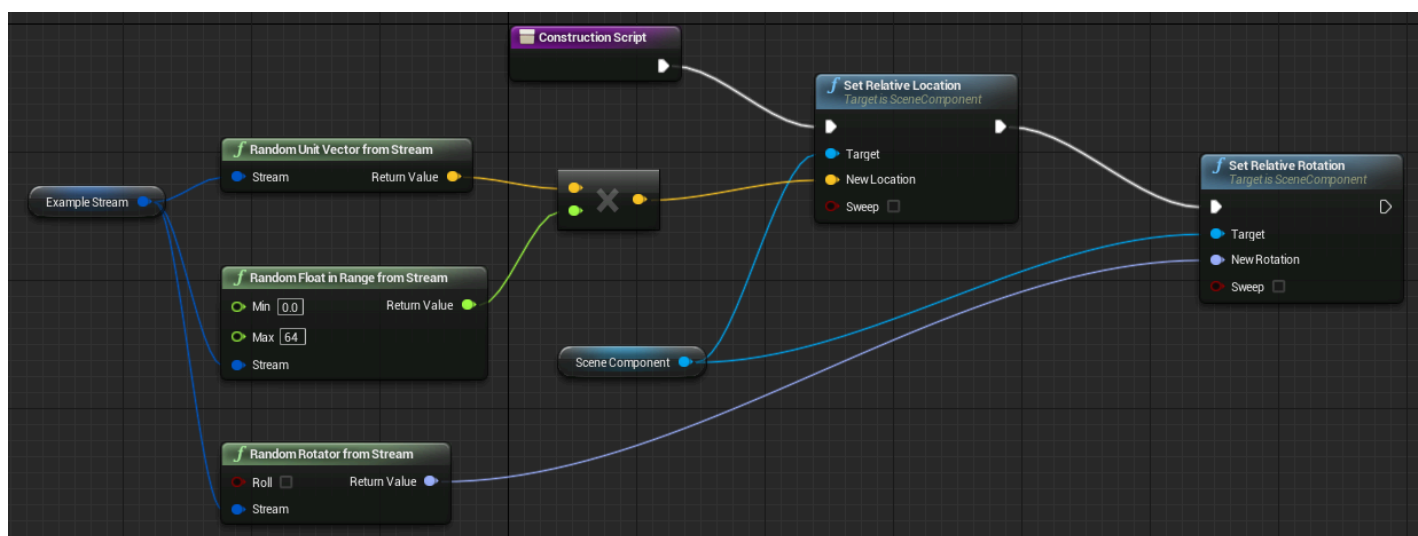
As you can see, random Booleans, Floats, Integers, Vectors, and Rotators can all be pulled from the same stream.

Initial Seed

The **Initial Seed** property is used to calculate the stream of random values. The sequence of random values produced from a single seed value will always be the same each time it is calculated, giving us the consistency mentioned previously. Different seeds generate different sequences of values.

So, modifying the Initial Seed of a RandomStream will cause the values produced to vary. This can be used to tweak a random effect until you get the sequence or distribution that suits your needs. A new Initial Seed is generated per-instance of the Blueprint. This means each time an instance of the Blueprint is placed in the world or a duplicate of an existing instance

is made, each RandomStream variable belonging to the Blueprint is given a new Initial Seed value. So, each instance will be different and can be tweaked to create the exact desired result.

Modifying the Initial Seed

To set the Initial Seed property directly on the variable, the RandomStream variable must be exposed for editing. Once exposed, the **Initial Seed** property is visible in the **Details** tab when an instance of the Blueprint is selected.

Within a Blueprint graph, you can also set the Initial Seed to a specific value or a new random value.

The **Set Random Stream Seed** node can be useful when you have one Blueprint that places many objects, like blades of grass or stones, but you want them to always be placed the same way for testing or other purposes. It will override the Initial Seed being randomly set when the Blueprint is placed in a level.