

Gameplay Blackboard Plugin

The Gameplay Blackboard Plugin is a flexible, replicated alternative to Unreal's default blackboard system. On any actor, it allows you to store any type of value, including custom structs and arrays, using Gameplay Tags as keys with full support for network replication.

In addition to per actor blackboards, it provides a Global Gameplay Blackboard through a subsystem, making it easy to centralize shared gameplay data and use it as a Global Variable Container.

Beyond AI, it's perfect for managing systems like game flow, phase systems, player stats, objectives, mission states, and other gameplay systems that need a centralized and/or replicated way to store any type of data.

Help: louisgirard.games@gmail.com

Gameplay Blackboard

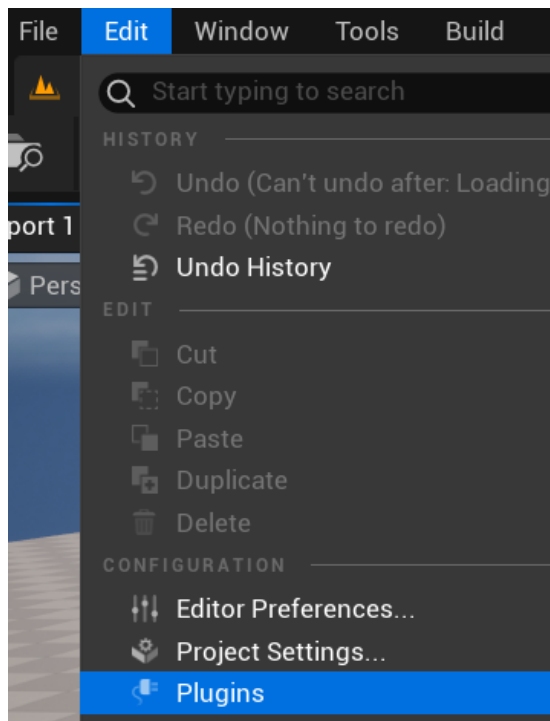


Store any type
Replicated

Activating the Plugin.....	3
Go to plugins.....	3
Enable the Gameplay Blackboard plugin.....	3
Gameplay Blackboard Component.....	4
Setters.....	4
Getters.....	6
On Value Changed.....	8
Global Gameplay Blackboard.....	10

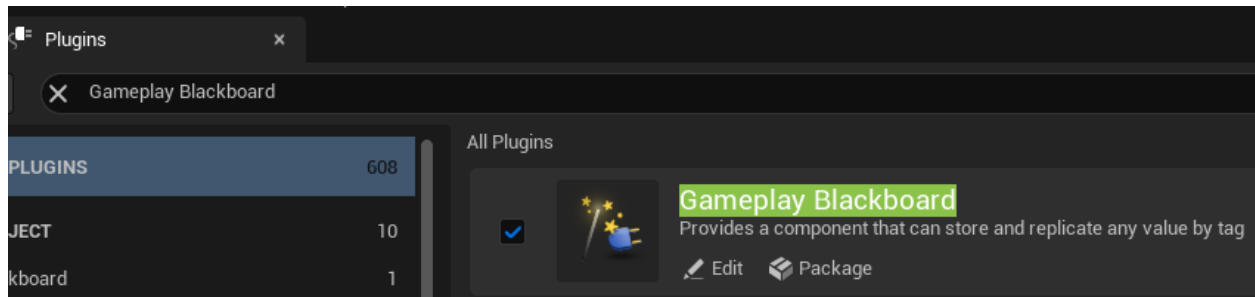
Activating the Plugin

Go to plugins



Enable the Gameplay Blackboard plugin

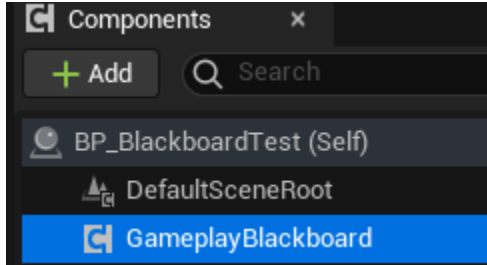
Search for Gameplay Blackboard and tick the box to enable it.



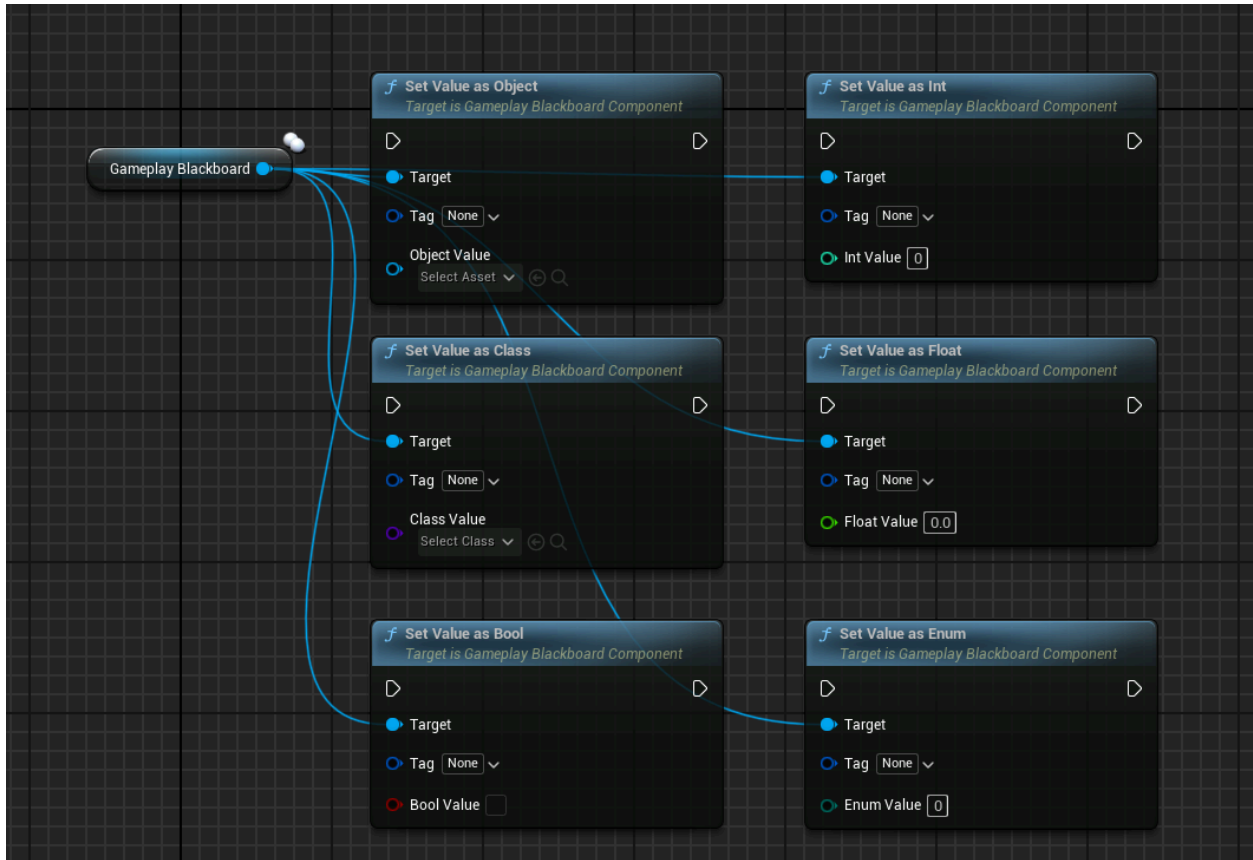
Gameplay Blackboard Component

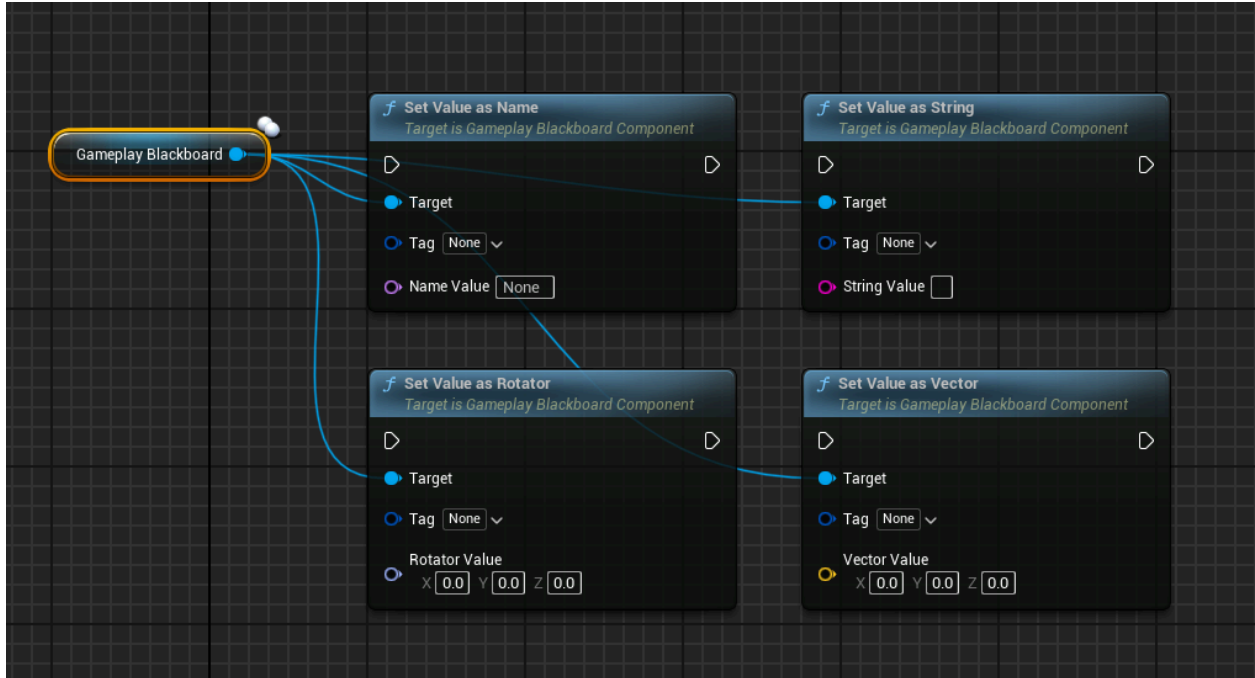
This is the core component. It will allow you to store any value of any type by key tags, and replicate those values.

Let's add this component to any test actor.



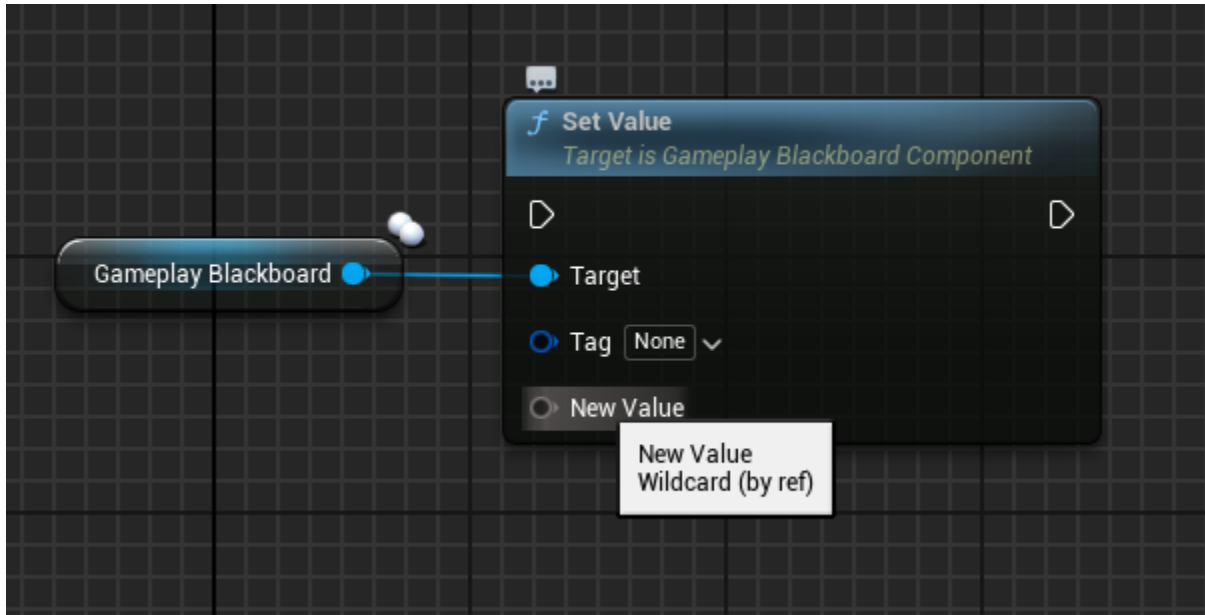
Setters



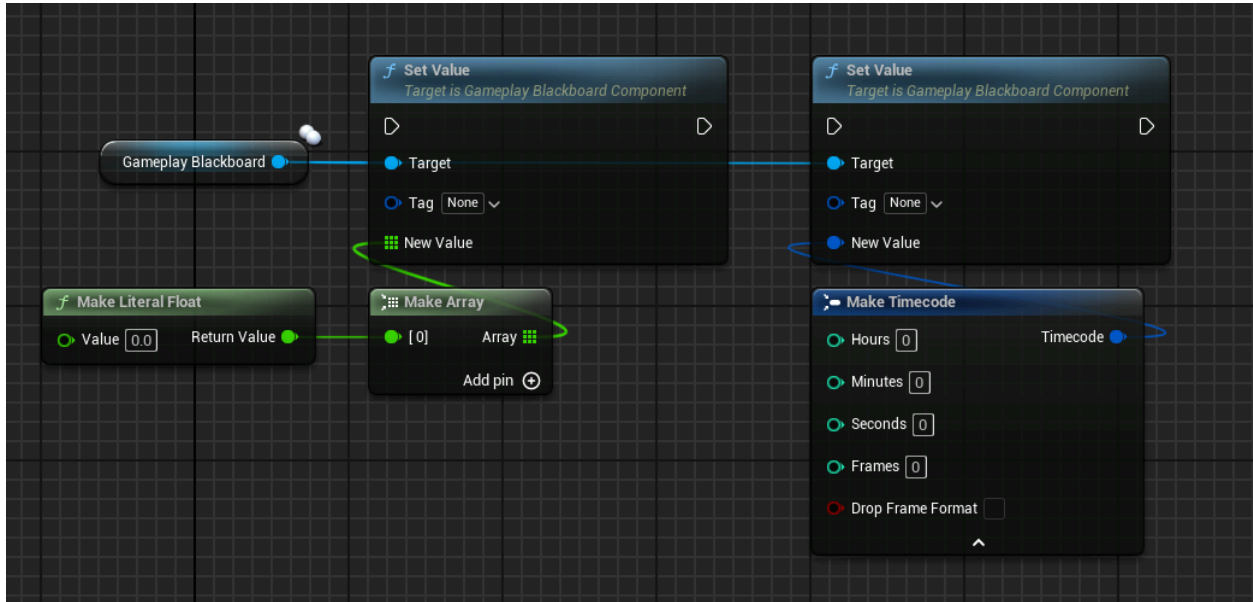


Like the default Unreal's Blackboard, you can access setters for some primitive types. You will always have to set a [Gameplay Tag](#) as a key to identify your value and get it later.

If you have any other type to set you can use the generic Set Value function that takes a Wildcard as input.

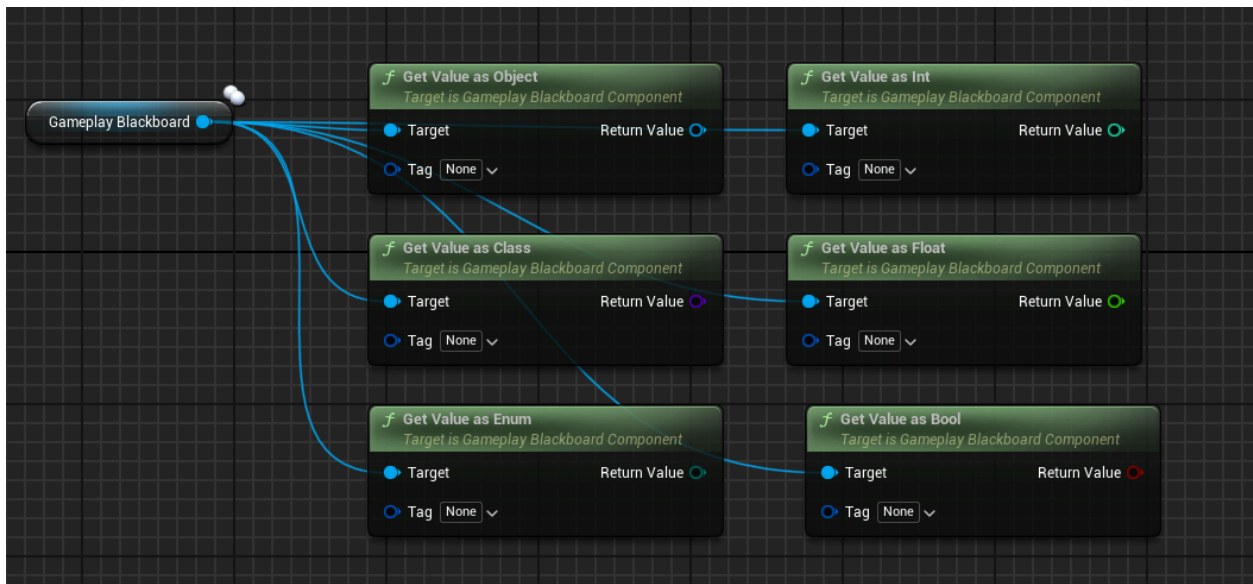


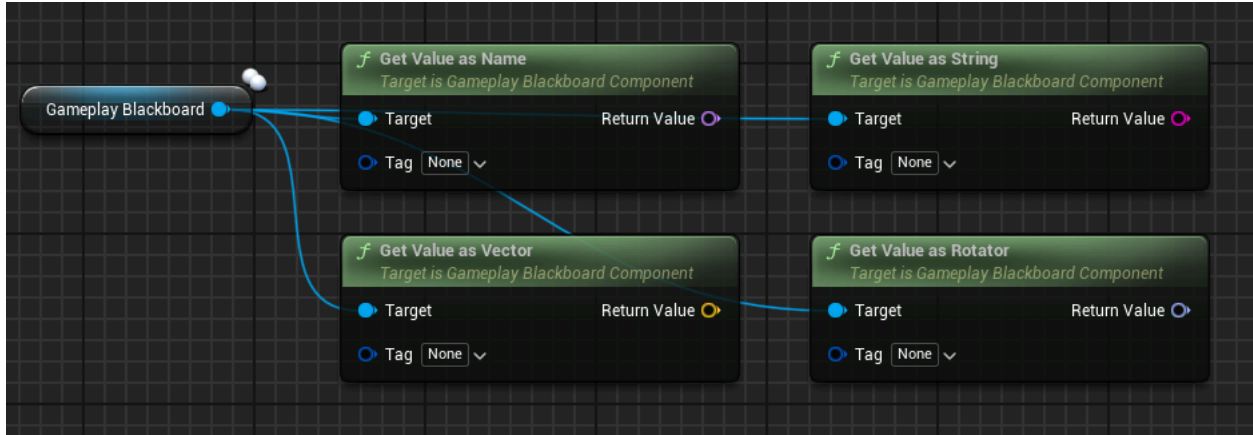
A **Wildcard** is basically any type, and waits for the user to plug something in to identify what type to convert to. You could for example store a customer Struct, or even an Array.



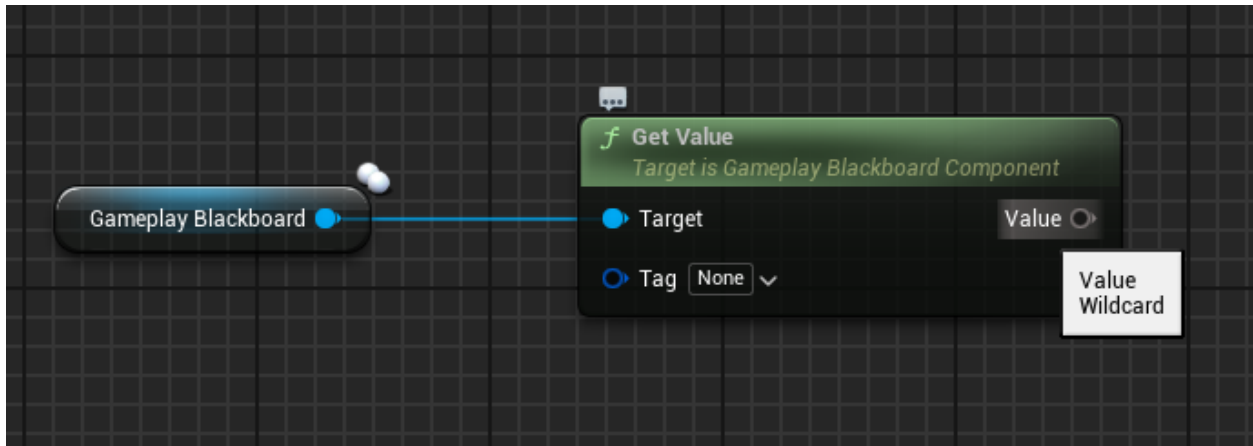
Getters

You have access to the getters for the same types as setters.

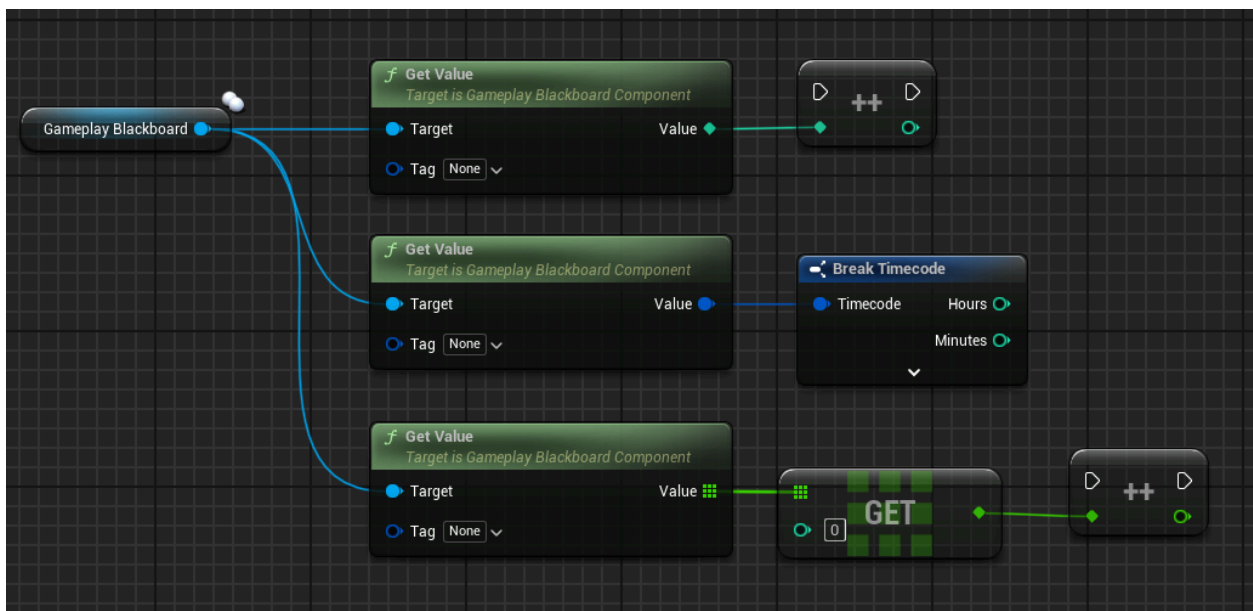




If you want to get any other type of values you can use the generic Get Value.



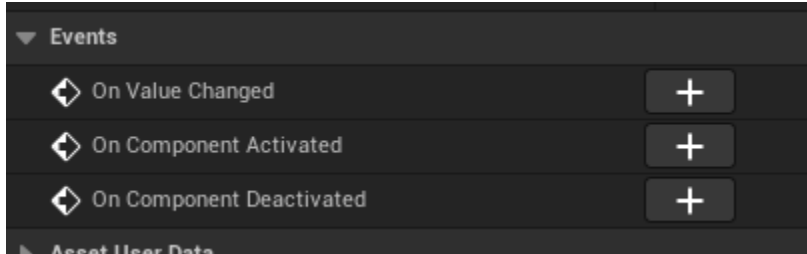
To convert the Value pin to its proper value you will need to use another node that takes that type of value. For example for an Integer a + operation, for a Struct a Break, for an Array a Get, etc..



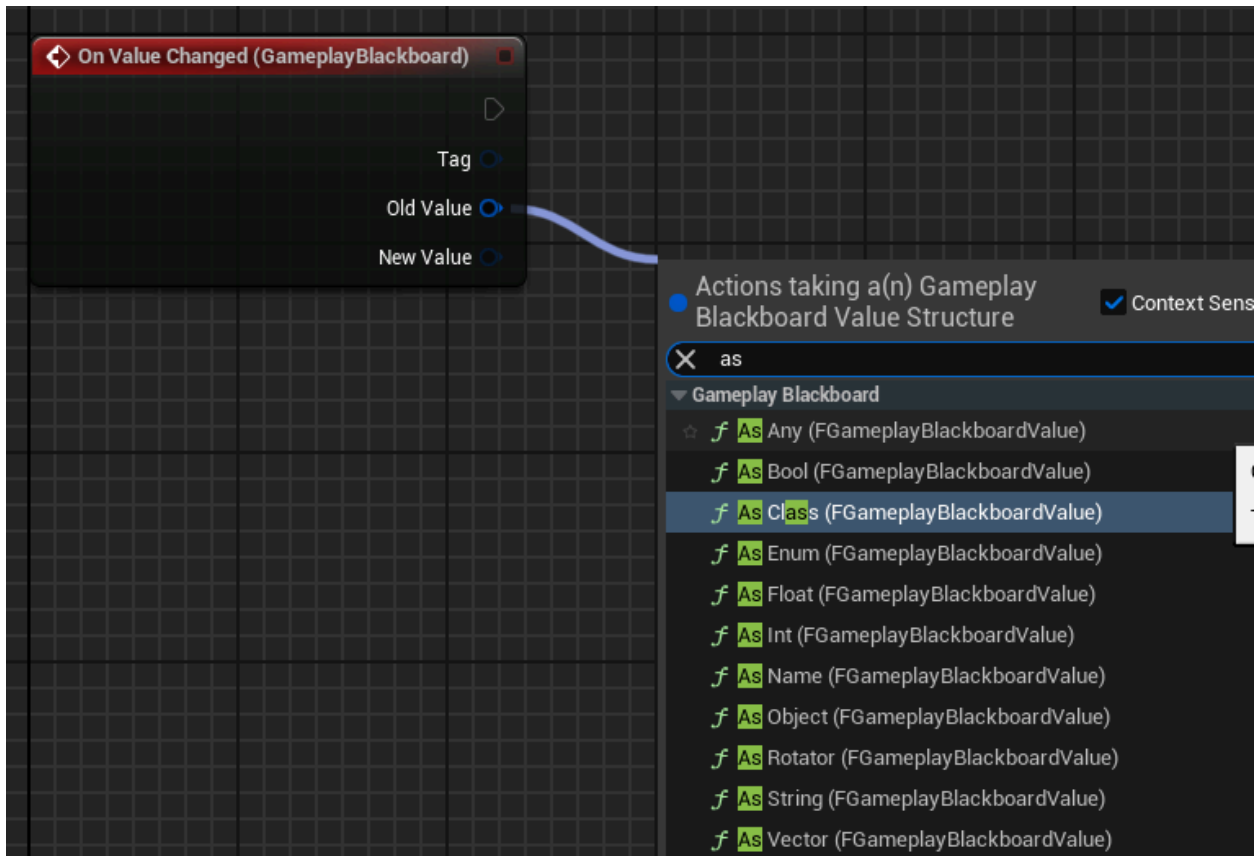
Important: When getting a value you should ensure you are getting a value with the same type that you set it, to avoid having a wrong value. For example if you set a String you should get the value as String and not as a Bool.

On Value Changed

On the component you have access to an event when a value is changed (added, removed, or updated).

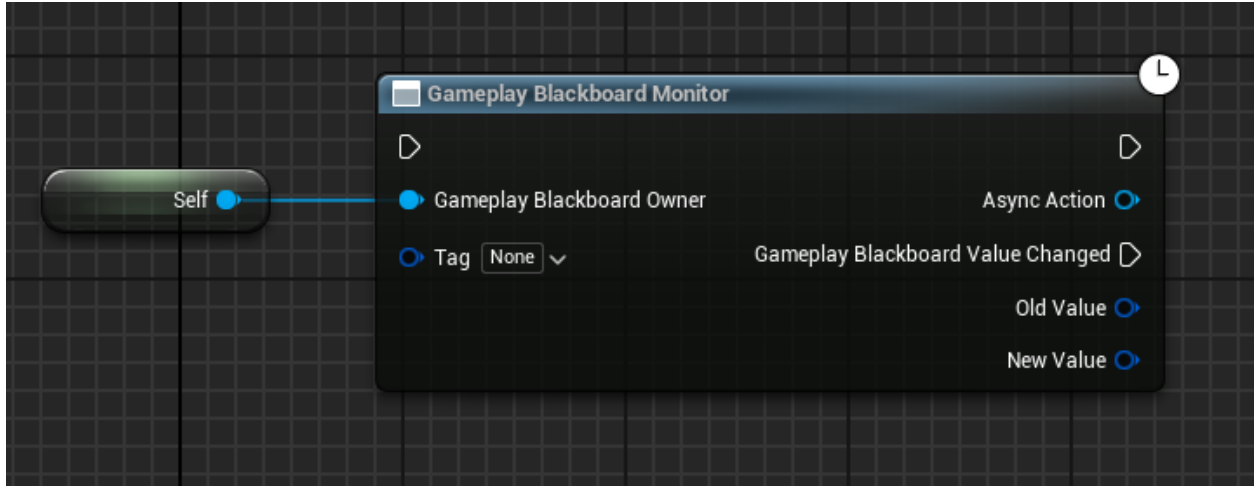


It will give you the Tag, the Old Value and the New Value. To read the Old and New Values you need to convert them as the type you want.



Important: Same for the getters, make sure you convert to the proper type, and to convert Old and New Value to the same type.

If you want to monitor only a change on a specific entry, you can use the Gameplay Blackboard Monitor.

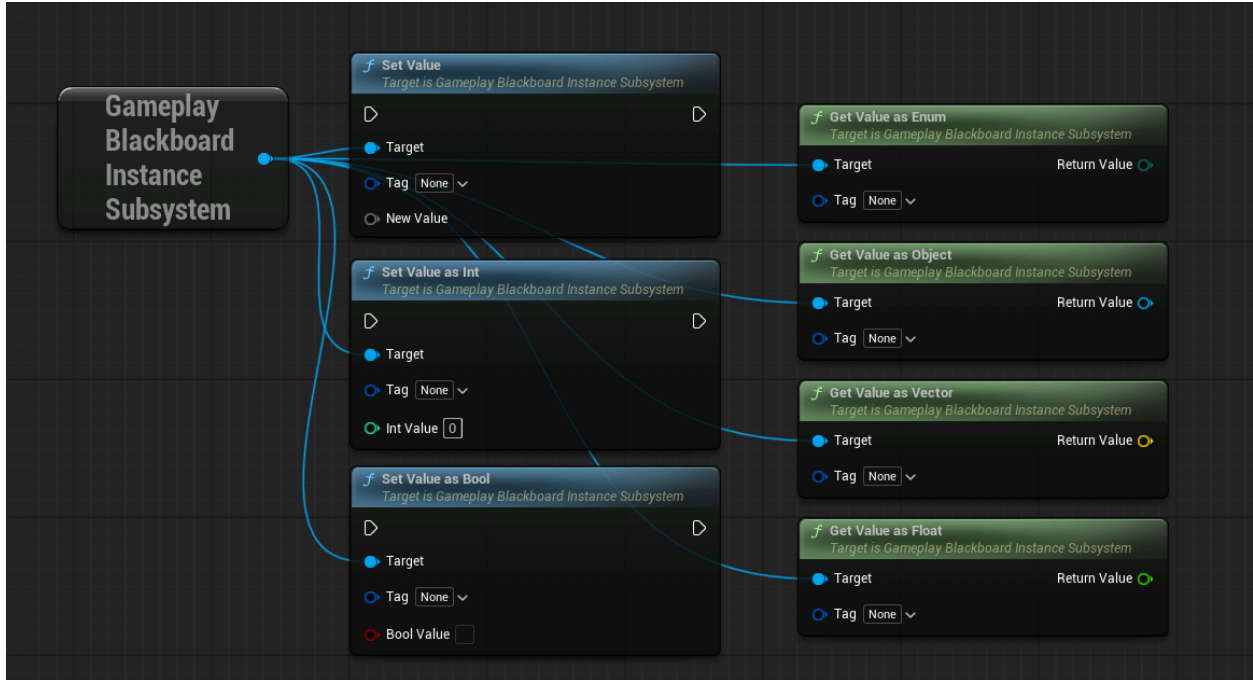


You specify the Tag and then you get the Old and New Value to convert exactly like the On Value Changed Event.

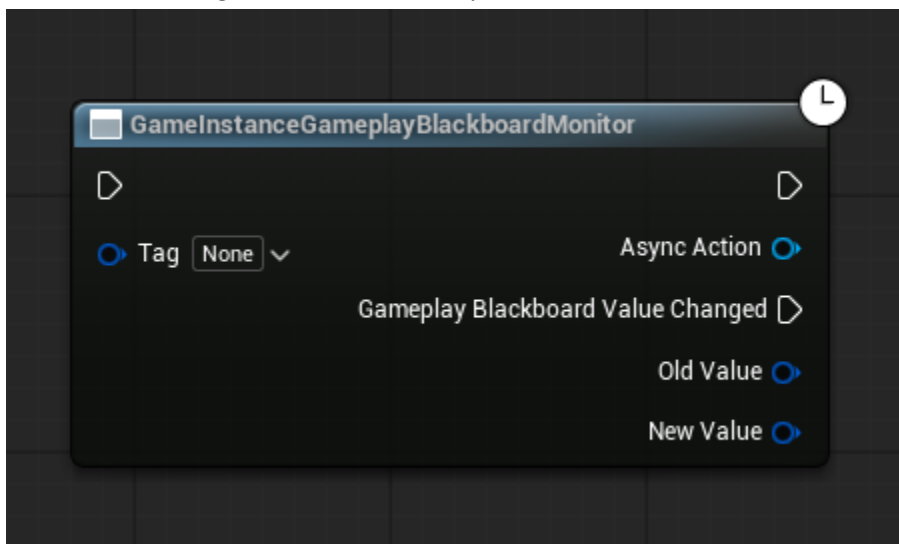
Global Gameplay Blackboard

If you need a Gameplay Blackboard that you can access from anywhere, without needing a Component, and to basically act as a Global Variables container, you can use the **Gameplay Blackboard Instance Subsystem**.

You will find the exact same functions as the **Gameplay Blackboard Component**, to set or get any value, the only difference being that these values won't be replicated since they are living on a local Game Instance Subsystem.



To listen to a change in this blackboard you can use the Game Instance Gameplay Blackboard Monitor.



With all that you should have all the information needed to start using the plugin and customize it the way you want. If you feel there is some information missing or you don't understand some part, feel free to contact me at louisgirard.games@gmail.com.