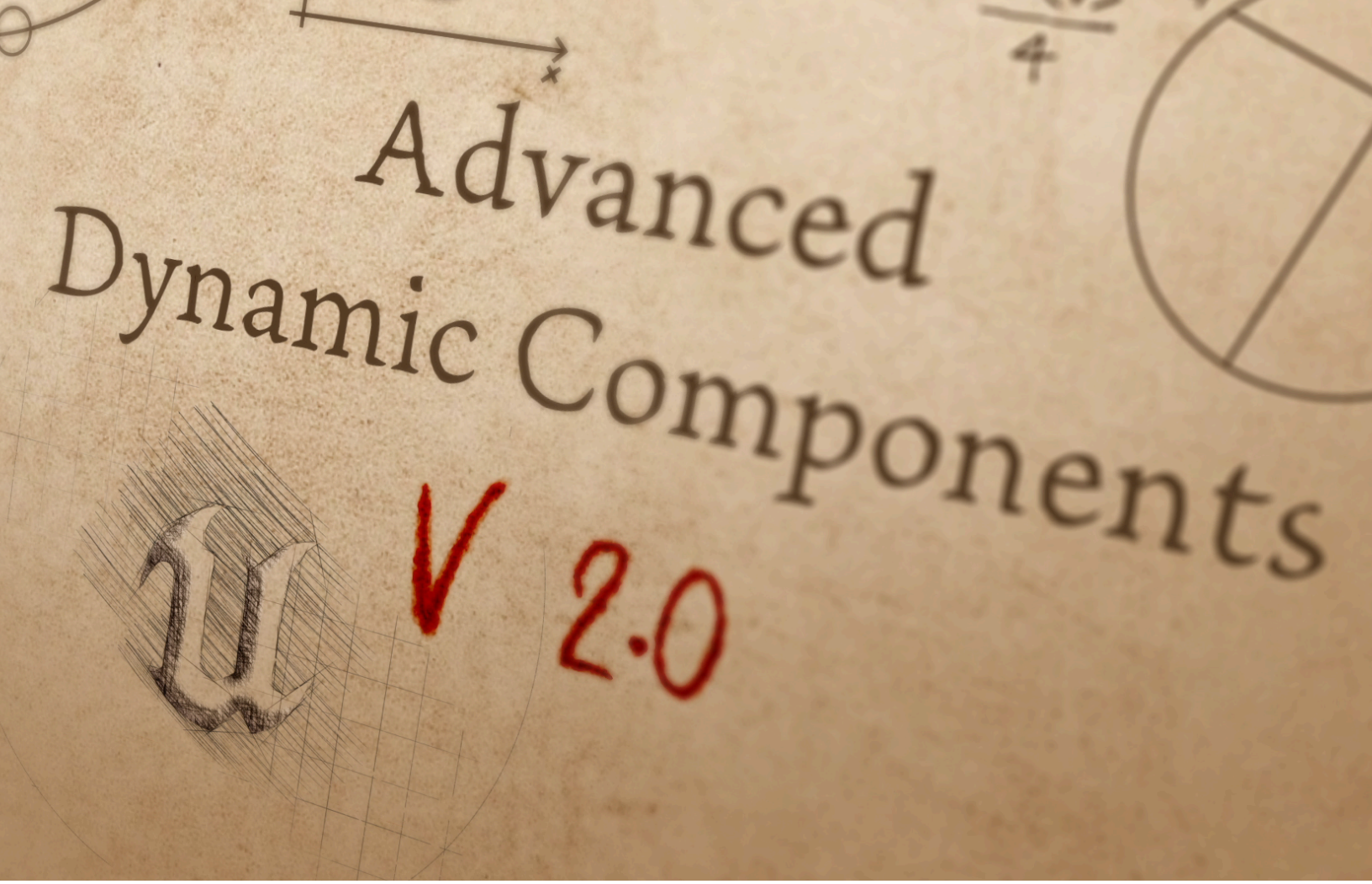


Welcome to Advanced Dynamic Components V2.0 documentation!

Paired animations, doors, health, stamina, status effects, temperature, sound, camera, light, interaction, examination, inventory, loot, equipment, materials, events system, spline and curve animation.

+

various materials, niagara emitters and common functions library

Index:

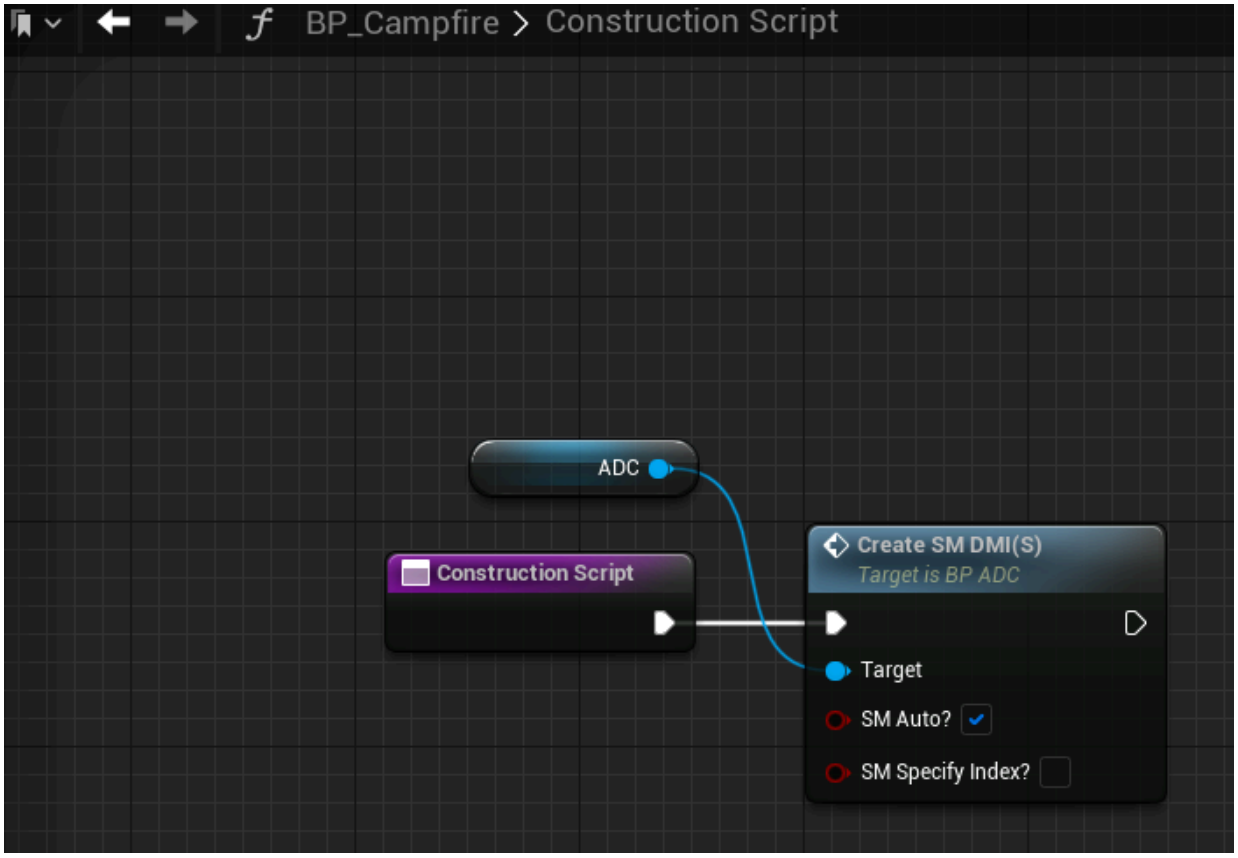
- *1. Dynamic **Material, light and anim** BP ADC*
- *2. Dynamic **Metasounds** - BP ADC SFX*
- *3. Dynamic **Camera** - BP ADC DCS*
- *4. Dynamic **Temperature** - BP ADC DTS*
- *5. Dynamic **Breathing** - BP ADC DPS*
- *6. Dynamic **Interaction** - BP ADC DIS*
- *7. Dynamic **Storage** - BP ADC DSS*
- *8. Dynamic **Equipment** - BP ADC DES*
- *9. Dynamic **Doors / Windows** - BP Anim Interact*
- *10. Dynamic **Paired Animations -** BP PairedAnims*
- *11. Dynamic **Vigor** - BP ADC DPS*
- *12. Dynamic **Status Effects** - BP ADC DPS*
- *13. Common **Functions Library** - BPF_ADC*
- *14. Roadmap*

Dynamic Materials, light & anim

to create dynamic materials using ADC, first we need to add ADC component to our actor blueprint by opening our blueprint then navigating to the + Add green button in the top left corner under the components section and once clicked we type in “ADC” and select it so it may be added to our blueprint.

Now that our component is added we can navigate to the **Construction Script** in the Functions tab found under My Blueprint section, double click on it to open its graph and now we can select our ADC component from the Components section and LMB drag it into the graph then we will LMB drag from BP_ADC and type in **“Create”** you have 4 options of dynamic materials, skeletal mesh, static mesh, niagara and Geometry Collection / Chaos and the process is the same for all.

assuming you have a static mesh component, we will select **“Create SM DMI(s)”** and connect it to our construction script



The two booleans **SM Auto** **&** **SM Specify Index** function as follows:

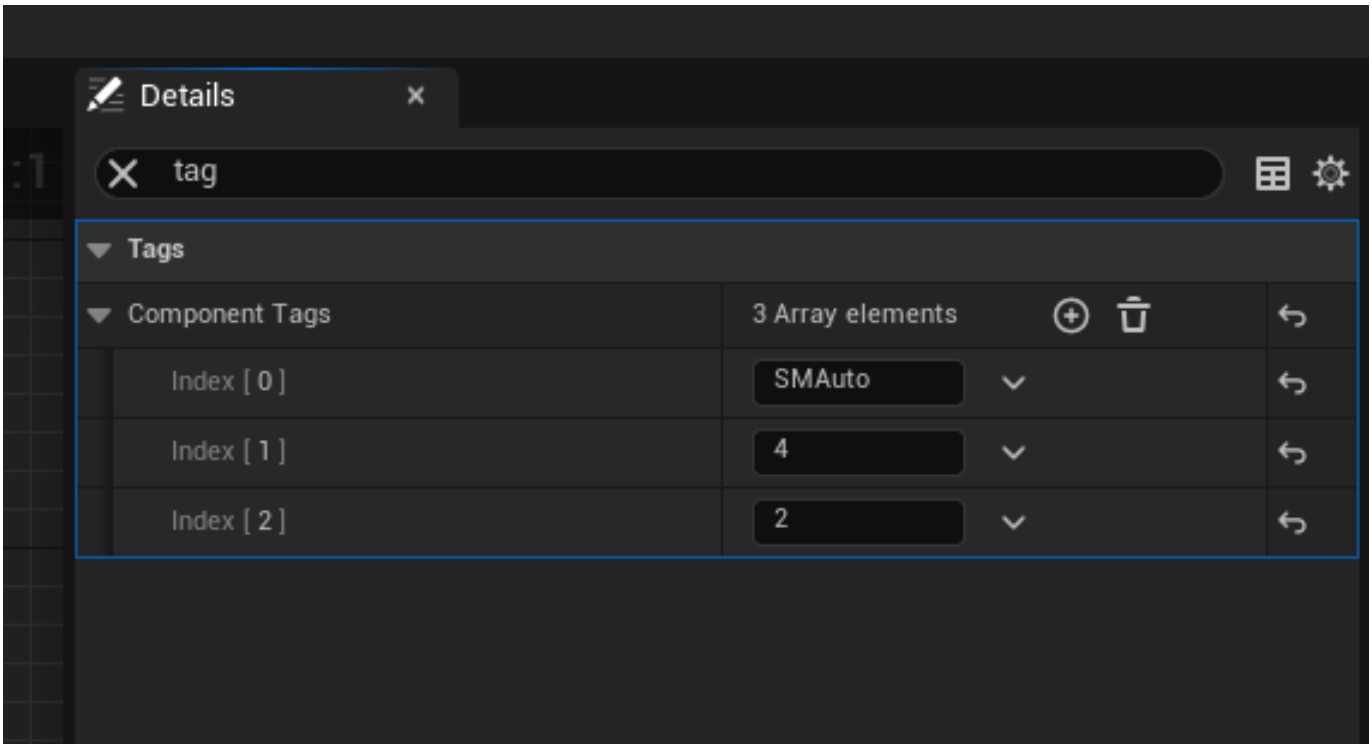
SM Auto: Detect all static mesh components and convert all of their materials to dynamic.

SM Specify Index: Detect component numerical tags (IE 4, 2) and use those numbers to convert only those material indexes (IE our material index 4 and 2 are the only ones dynamic now)

If you turn off “SM Auto” you will need to find your static mesh component(s) you want to convert and add a new component tag it/them **“SMAuto”** and it will convert all of its material indexes to dynamic.

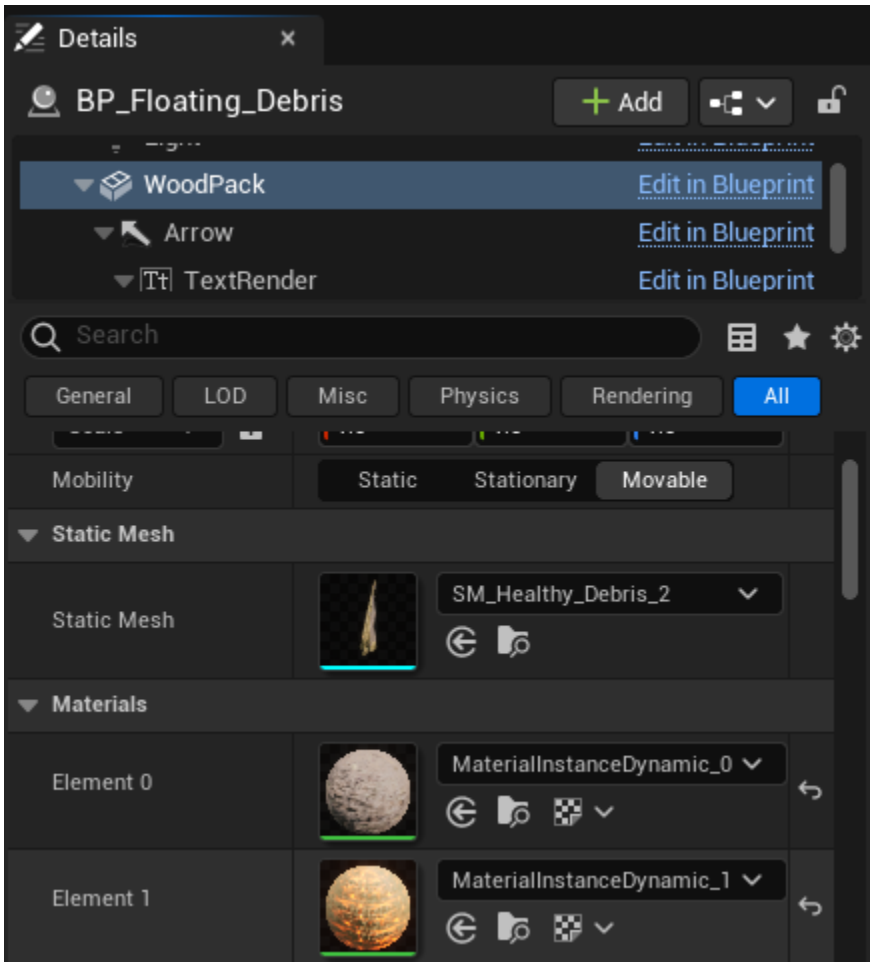
If you turn on “SM Specify Index” you will need to add the material indexes to the component tags that you want to be converted to dynamic.

> **IMPORTANT**: if adding and specifying index, make sure your numerical values ARE NOT THE FIRST INDEX so instead position your **SMAuto** tag as follows:



Now you can drag in your blueprint to the level, navigate to its static mesh and find its materials to find that our materials have been converted to dynamic.

If you have been following correctly, your materials tab will look like this:

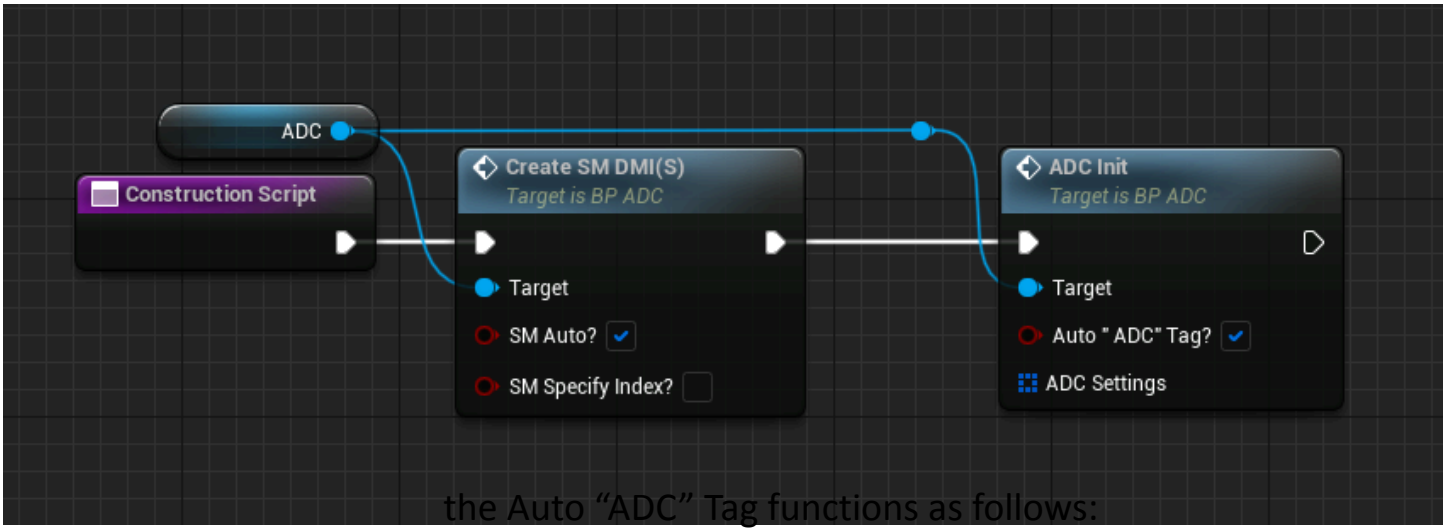


==========

Next we will edit our material parameters using ADC.

-Editing Dynamic Materials -

To start editing our newly created dynamic materials we will need to head back to our construction script, find our ADC that we placed into it and LMB drag from it and type in “ADC Init” and select it.



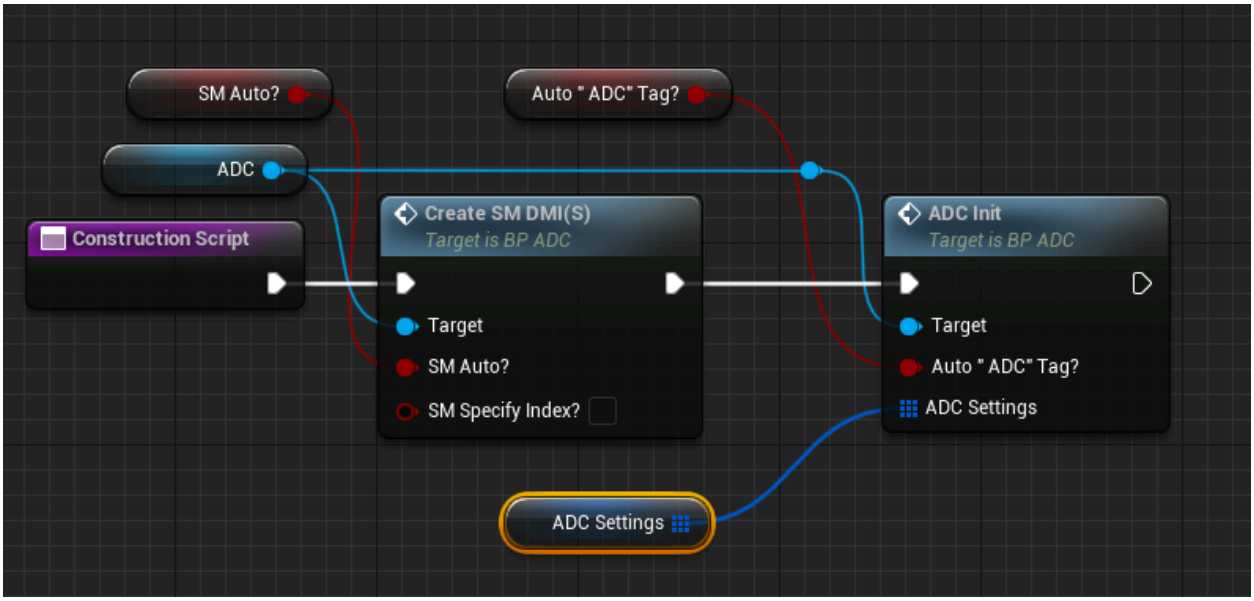
the Auto “ADC” Tag functions as follows:

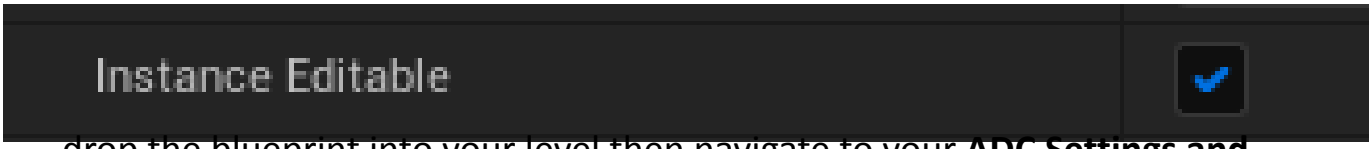
Auto “ADC” Tag: TRUE enable editing on all dynamic materials, light and Niagara components

If Auto “ADC” Tag is set to false you will need to find your dynamic component(s) and give them the “ADC” tag to enable editing on them. Same with Niagara, Light, Audio, SKMs and Chaos component

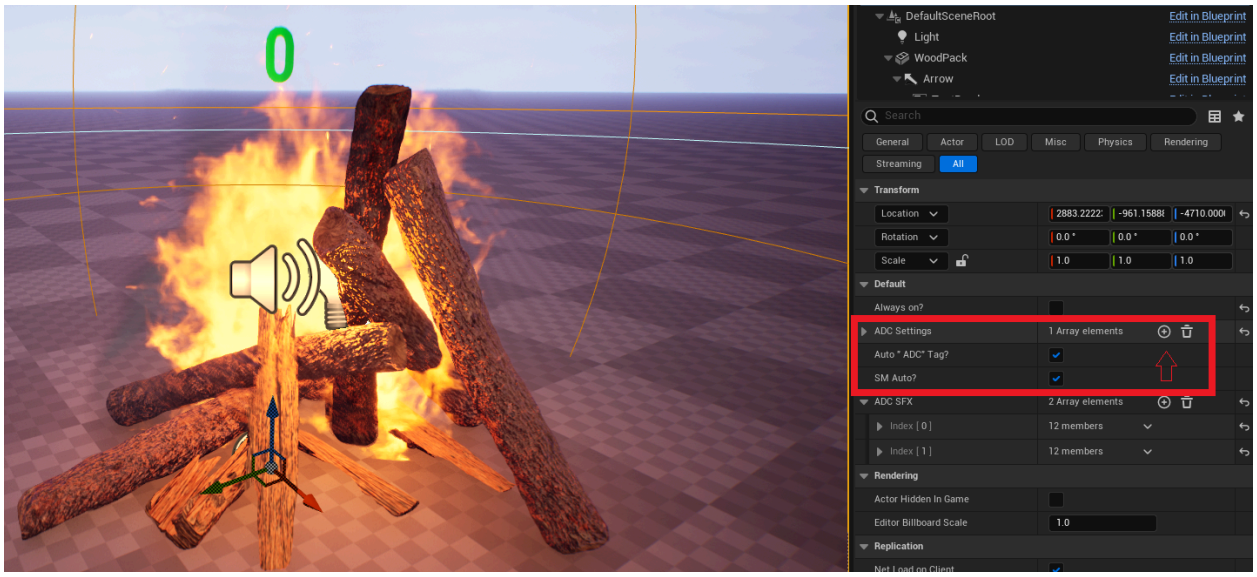
Next we need to promote ADC Settings to variable, so we right click on it then select promote to variable , after that we make it instance

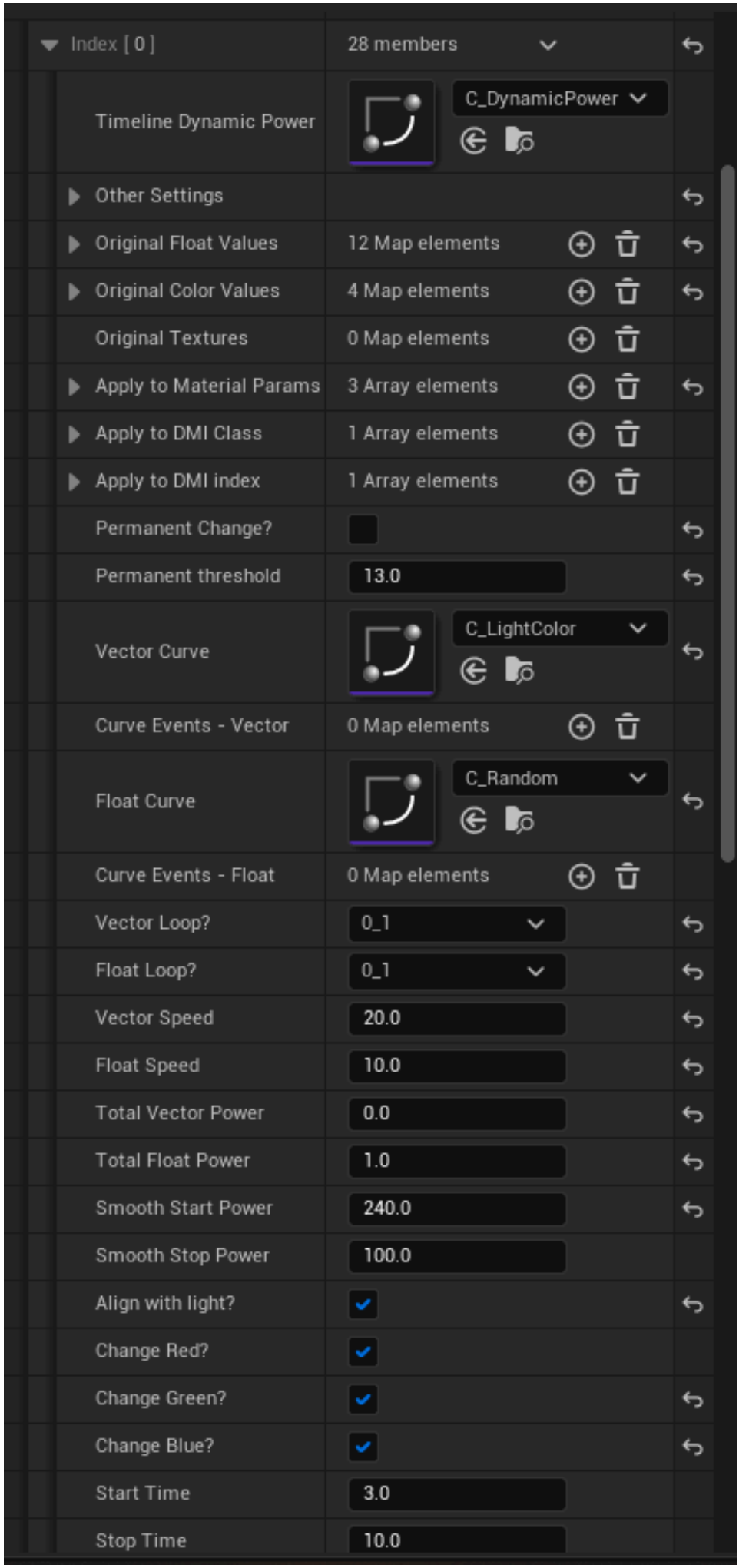
edible through Details tab on the right so we can use the settings in the level.





drop the blueprint into your level then navigate to your **ADC Settings and add an array element and we will go through each settings.**



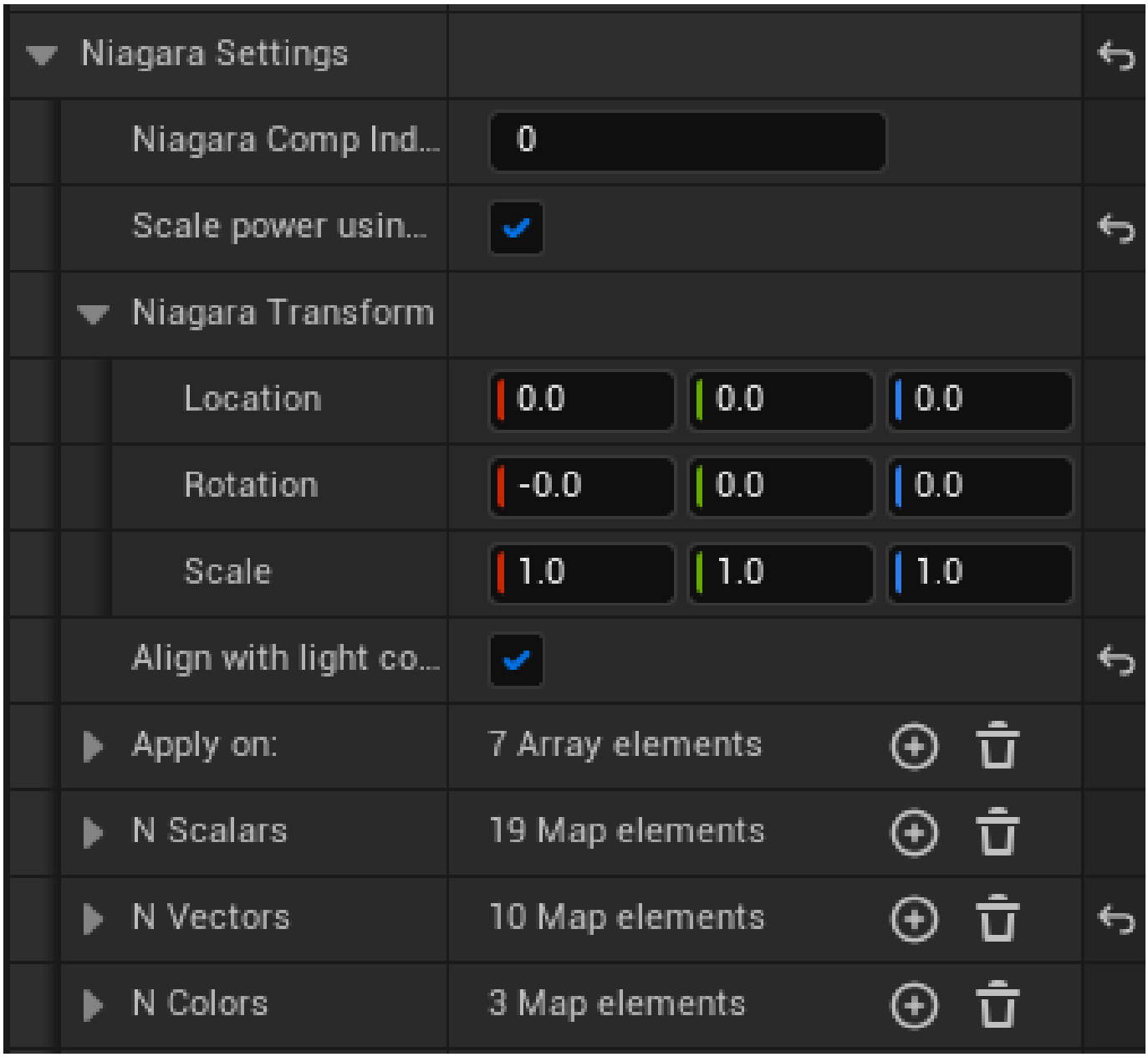

Index [0]
28 members
Timeline Dynamic Power
C_DynamicPower
Other Settings
Original Float Values
12 Map elements
Original Color Values
4 Map elements
Original Textures
0 Map elements
Apply to Material Params
3 Array elements
Apply to DMI Class
1 Array elements
Apply to DMI index
1 Array elements
Permanent Change?
Permanent threshold
13.0
Vector Curve
C_LightColor
Curve Events - Vector
0 Map elements
Float Curve
C_Random
Curve Events - Float
0 Map elements
Vector Loop?
0_1
Float Loop?
0_1
Vector Speed
20.0
Float Speed
10.0
Total Vector Power
0.0
Total Float Power
1.0
Smooth Start Power
240.0
Smooth Stop Power
100.0
Align with light?
Change Red?
Change Green?
Change Blue?
Start Time
3.0
Stop Time
10.0

Timeline Dynamic Power: used for time value and scaling niagara values through its power.

Other Settings: Used for niagara and comp animation system via curves and splines, when you want to animate via spline add the spline to your BP add the **TAG** to the spline in respect of what component you want to animate;

Static Mesh = SM, Skeletal Mesh = SKM, Niagara = Niagara

Other Settings, Niagara:



Scale Power bool: when true will use the first curve for time value to scale its overall values by 1, if false will use material float curve

Transform: used for editing relative location, this is fired in editor time.

Apply on: select which parameters you want to scale when toggled

N Scalars, Vectors, colors: used for editing original values during editor time by inserting name then corresponding value.

==========

ADC Settings continued:

Original Float Values: insert name of parameter and value to edit in editor time.

Original Color Values: insert name of parameter and value to edit in editor time.

Original Textures: insert name of parameter and value to edit in editor time.

Apply to DMI Class: select which dynamic material to edit in editor time.

Apply to DMI Index: select which index of the material element to edit in editor time.

Apply to material params: select which material params to change using curve values.

Permanent change bool: will stop editing values at a given **threshold if true**.

Vector Curve: the vector curve for editing color values on the materials.

Float Curve: the float curve for editing scalar values on the materials and also niagara if bool is set to false.

Curve Events, float / vector: used for firing custom events when the selected curve reaches a certain time, best to use non decimal values.

Curve Speeds: if 0 will traverse at normal 1.0 time, use negative values to slow down the curve.

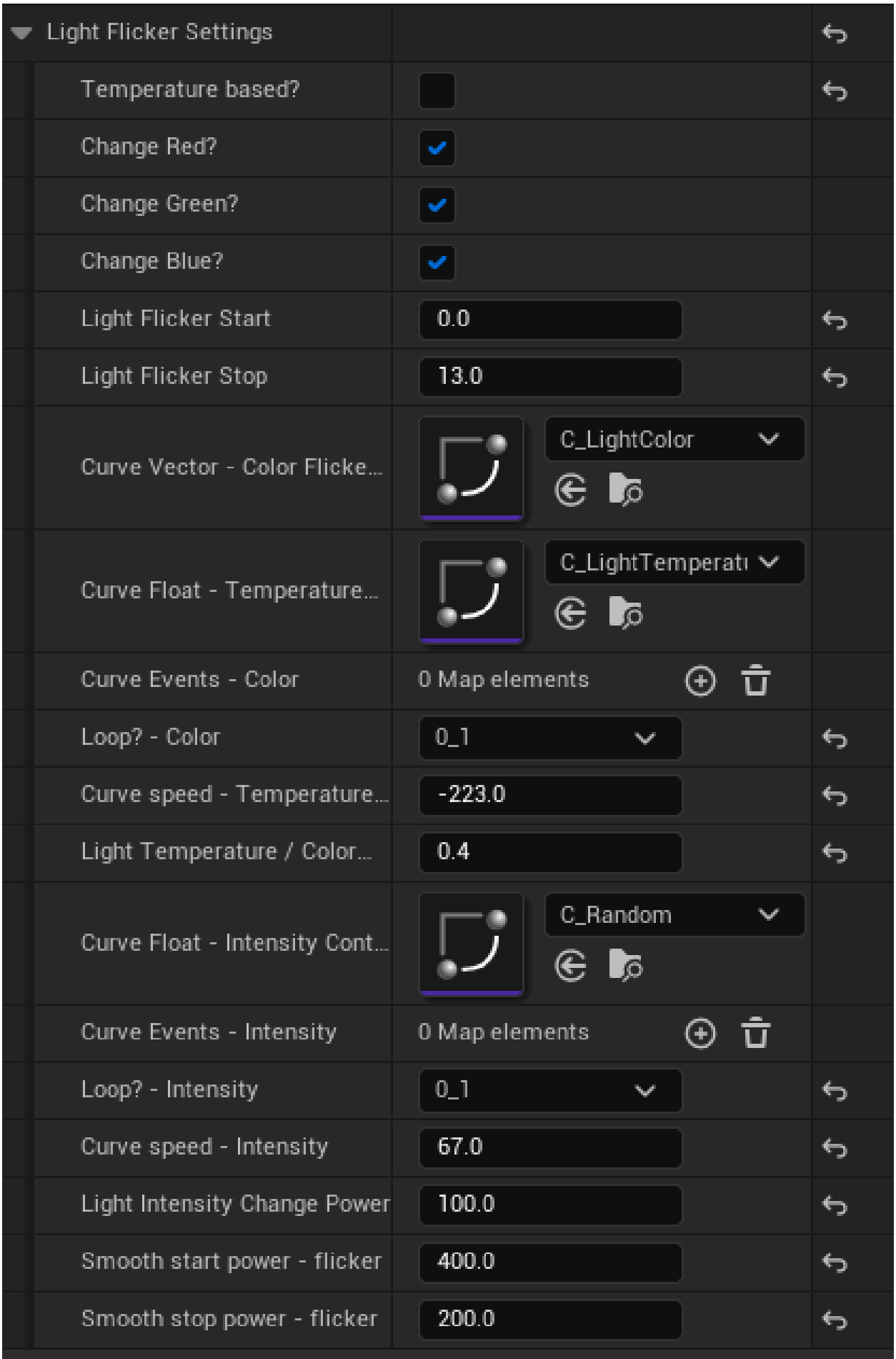
Vector / Float Loop?: looping method, 0_0 back to back, 0_1 back and forth or no looping.

Total Float / vector power: overall multiplier for applied values.

Smooth Powers: used to smooth toggling, higher values will be slow.

Start + Stop times: toggles change when the timeline dynamic power has reached the specified time.

Next we will look at the component in the detail tab where light manipulation takes place.


Light Flicker Settings
Temperature based?
Change Red?
Change Green?
Change Blue?
Light Flicker Start
0.0
Light Flicker Stop
13.0
Curve Vector - Color Flicke...
C_LightColor
Curve Float - Temperature...
C_LightTemperati
Curve Events - Color
0 Map elements
Loop? - Color
0_1
Curve speed - Temperature...
-223.0
Light Temperature / Color...
0.4
Curve Float - Intensity Cont...
C_Random
Curve Events - Intensity
0 Map elements
Loop? - Intensity
0_1
Curve speed - Intensity
67.0
Light Intensity Change Power
100.0
Smooth start power - flicker
400.0
Smooth stop power - flicker
200.0

Temperature based: if light has temp enabled it will manipulate the temperature rather than the color.

Start + Stop times: toggles change when the timeline dynamic power has reached the specified time.

Curve Events, float / vector: used for firing custom events when the selected curve reaches a certain time, best to use non decimal values.

Loop?: looping method, 0_0 back to back, 0_1 back and forth or no looping.

Curve Speeds: if 0 will traverse at normal 1.0 time, use negative values to slow down the curve.

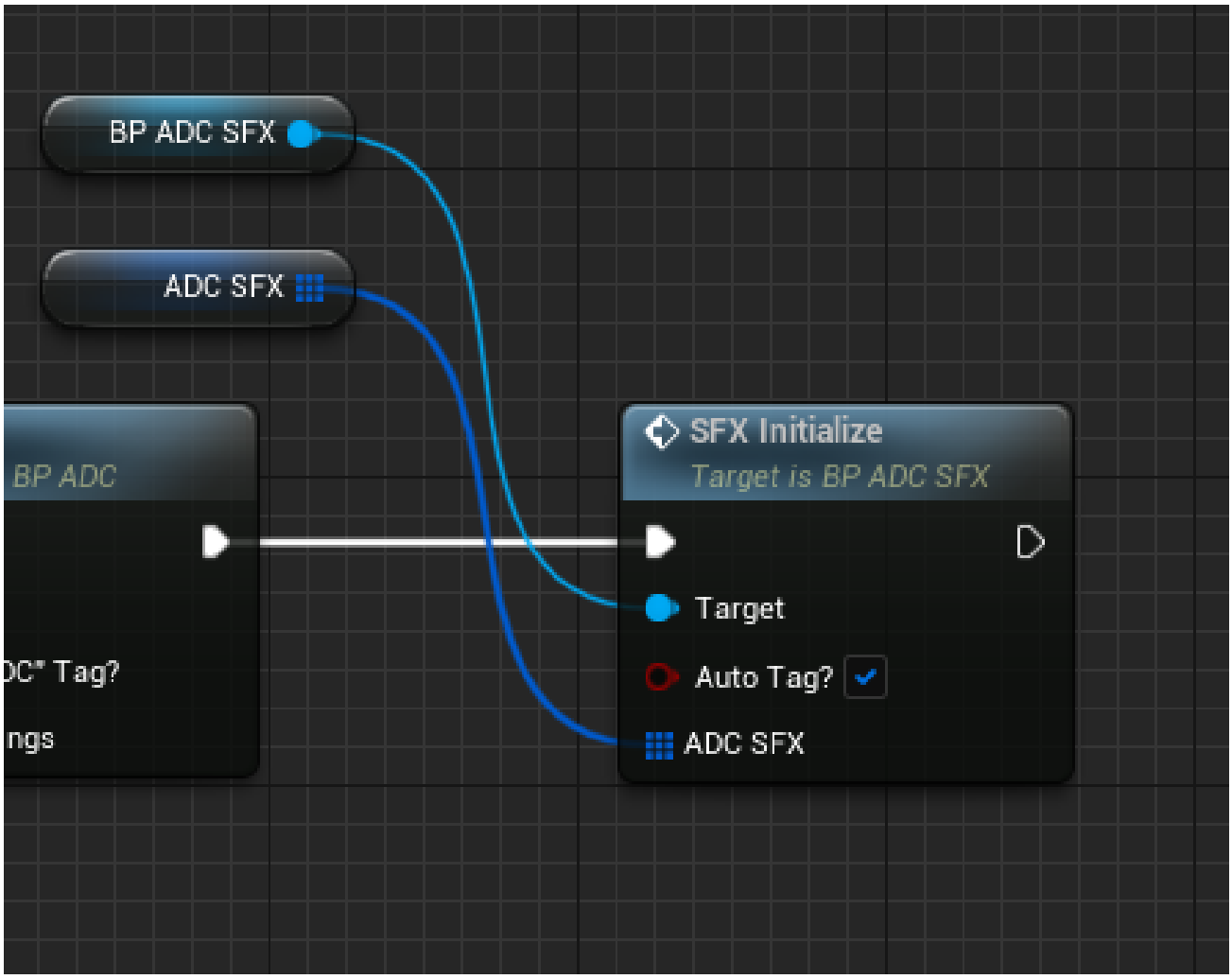
Change Powers: overall multiplier for applying value.

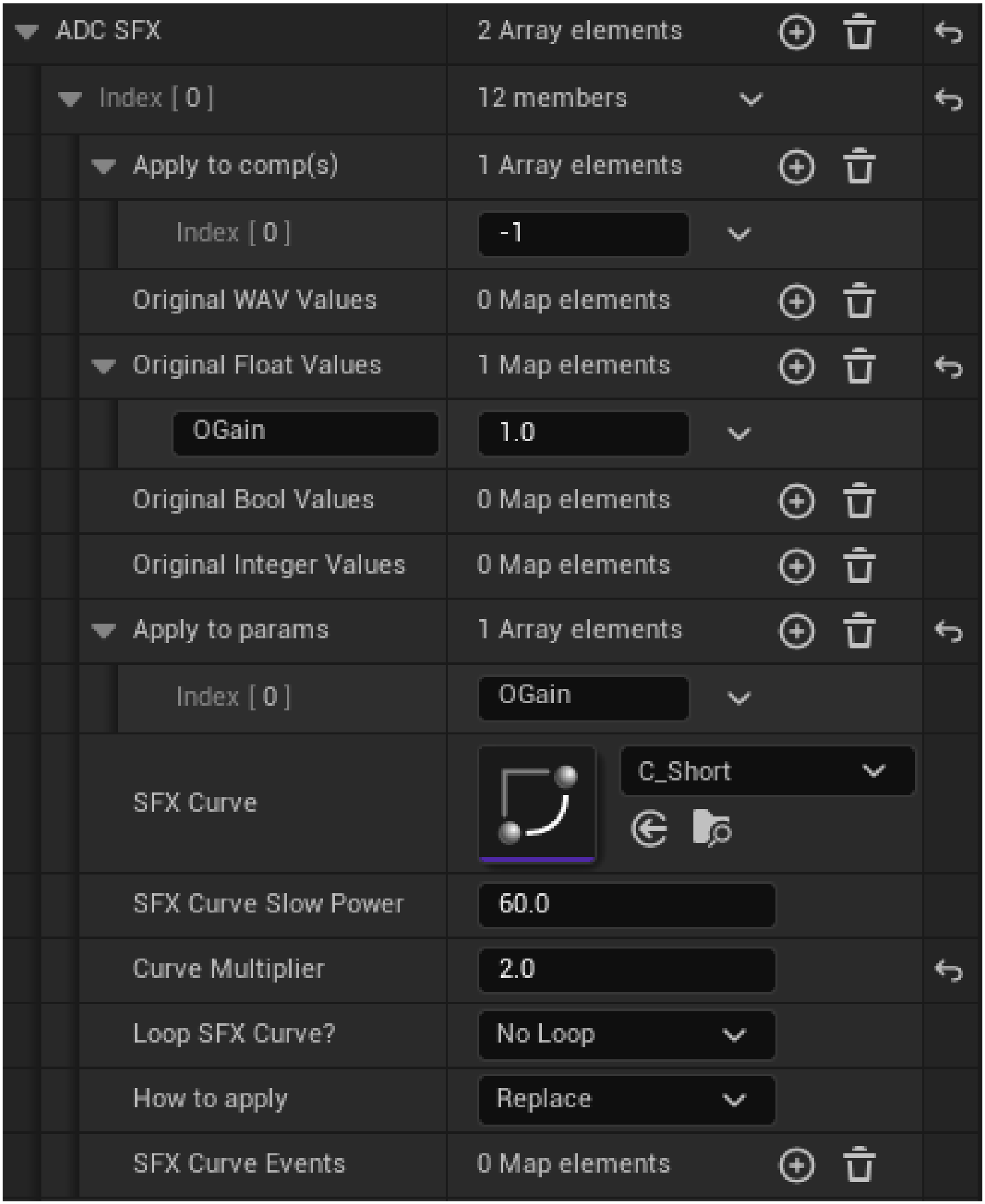
Useful variables: Anim toggle, ADC Time, Out Float, Out Color.

==========

Dynamic Metasounds

Initialize: follow the same procedure for ADC**.**



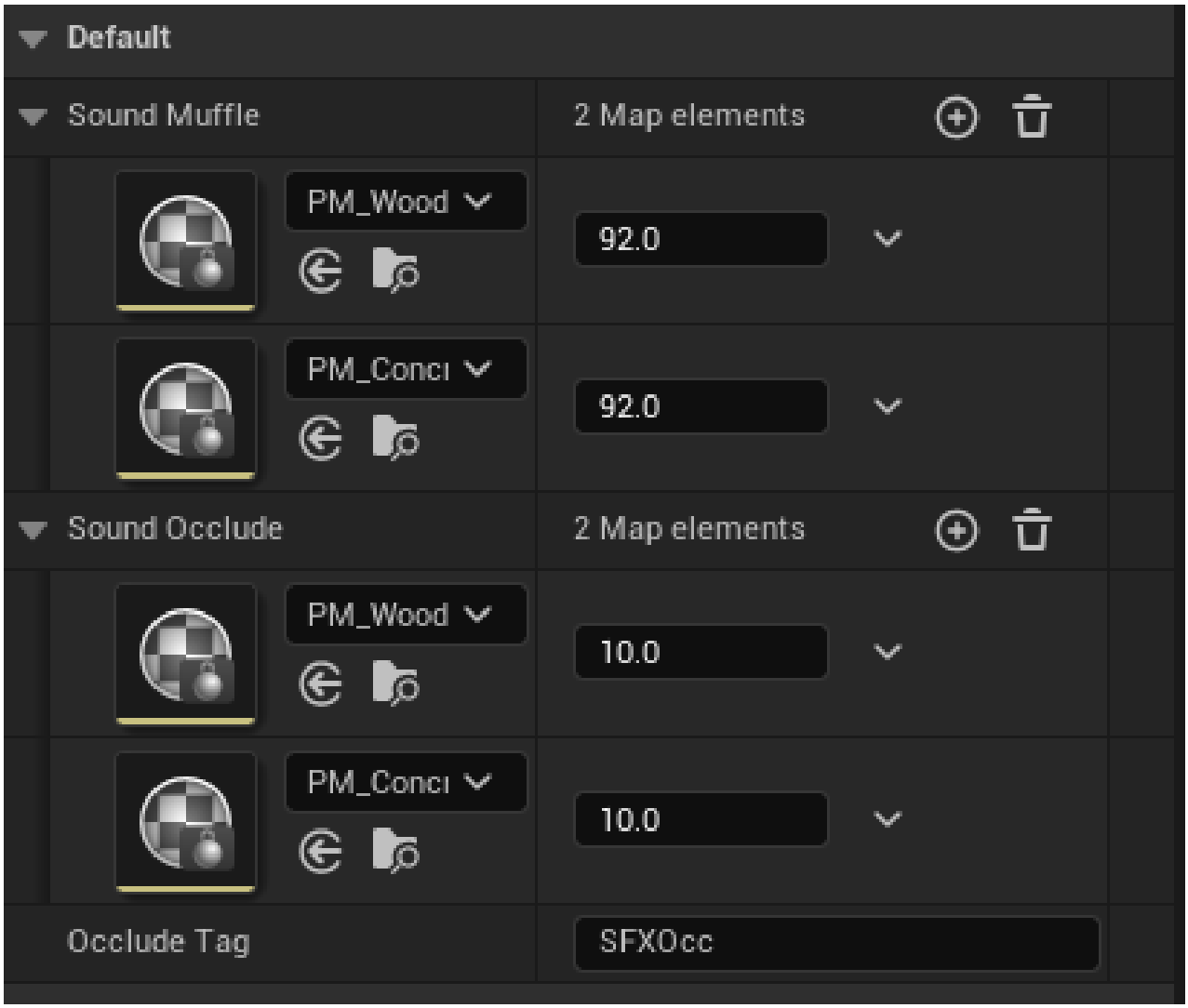

ADC SFX
2 Array elements
Index [0]
12 members
Apply to comp(s)
1 Array elements
Index [0]
-1
Original WAV Values
0 Map elements
Original Float Values
1 Map elements
OGain
1.0
Original Bool Values
0 Map elements
Original Integer Values
0 Map elements
Apply to params
1 Array elements
Index [0]
OGain
SFX Curve
C_Short
SFX Curve Slow Power
60.0
Curve Multiplier
2.0
Loop SFX Curve?
No Loop
How to apply
Replace
SFX Curve Events
0 Map elements

Original Values: select and add which input values to change, abehaves as a dynamic material instance.

Change Powers: select and type which material params to apply change to via the curve value.

SFX Curve slow power behaves the same as speed, just named differently: if 0 will traverse at normal time speed, use negative values for slowing down.

How to apply: Select which method of apply on the original values, divide, subtract, add or replace.



navigate to the component via details tab to find the settings.

Sound Muffle: add physical material and how much they should apply muffle effect in percentage (Will not reduce volume)

Sound Occlude: add physical material and how much they should reduce sound volume by in percentage.

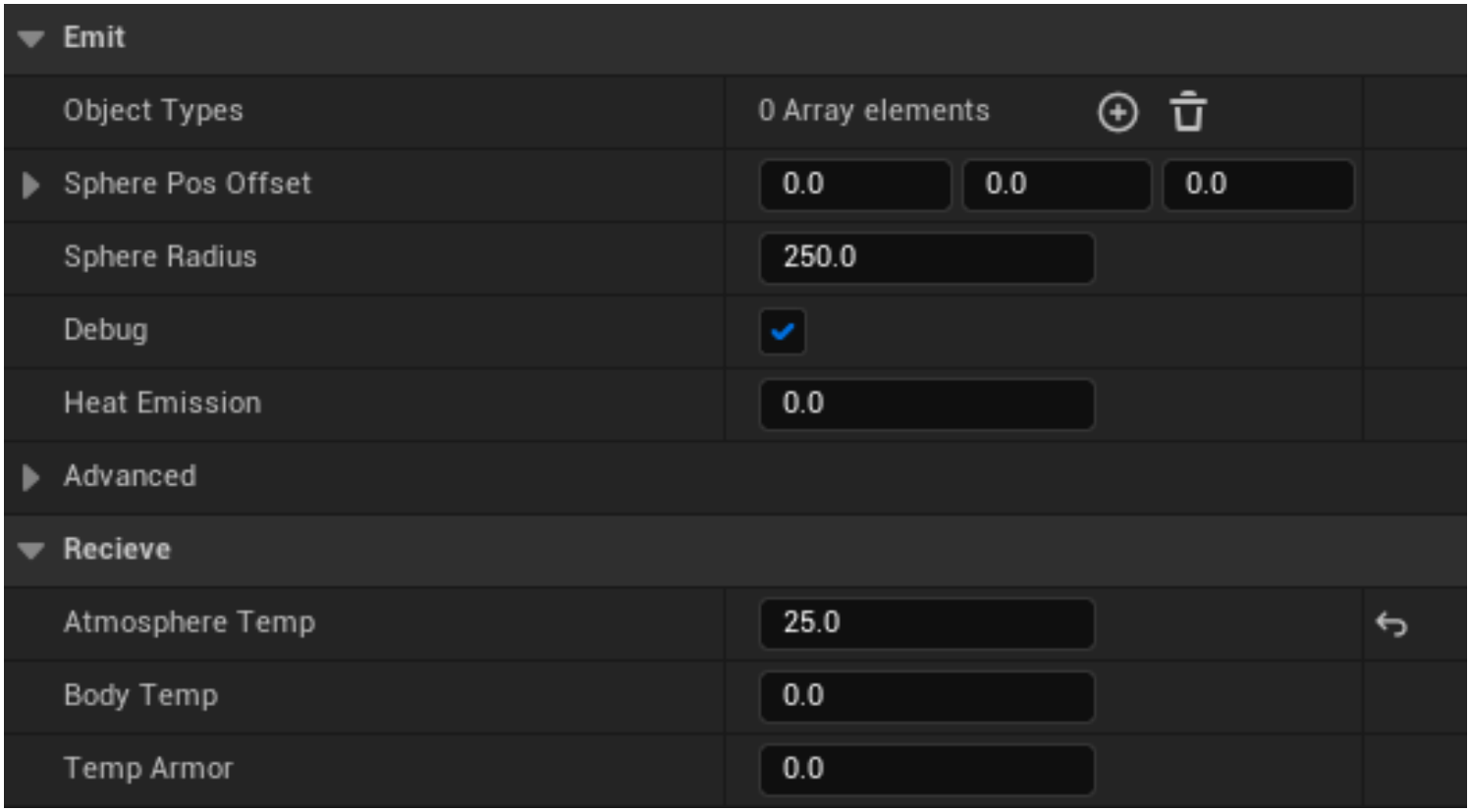
Occlude Tag: the tag that must be added to the sound components to apply effects.

==========

Dynamic Temperature System

EMIT: This section controls the actor which is responsible for emitting heat set by **Heat Emission** to different OBJECTS set by the **Object Types** in its radius set by the **Sphere Radius.**

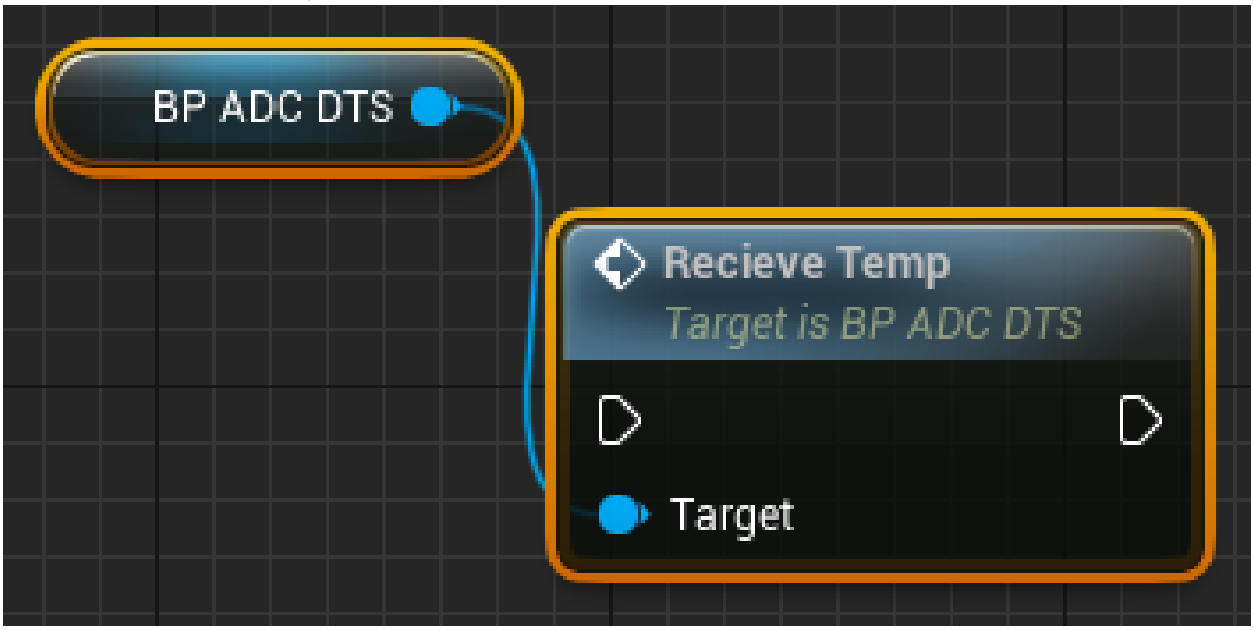
RECEIVE: This section controls the player/object;
Temperate Armor which controls how much heat is lost over time and increases heat gain when receiving greater than body temperature.
The **Atmosphere Temperature** is the default temperature of the current world, it will be applied on the player/object if they are NOT in the radius of any heat source.
Body Temperature should NOT be set directly, instead it should be a read only variable for other systems to work with such as DCS, HUD, Health and other systems.



TICK FOR OBJECTS RECEIVING TEMPERATURE:

Drag **BP_ADC_DTS** into the main graph and call

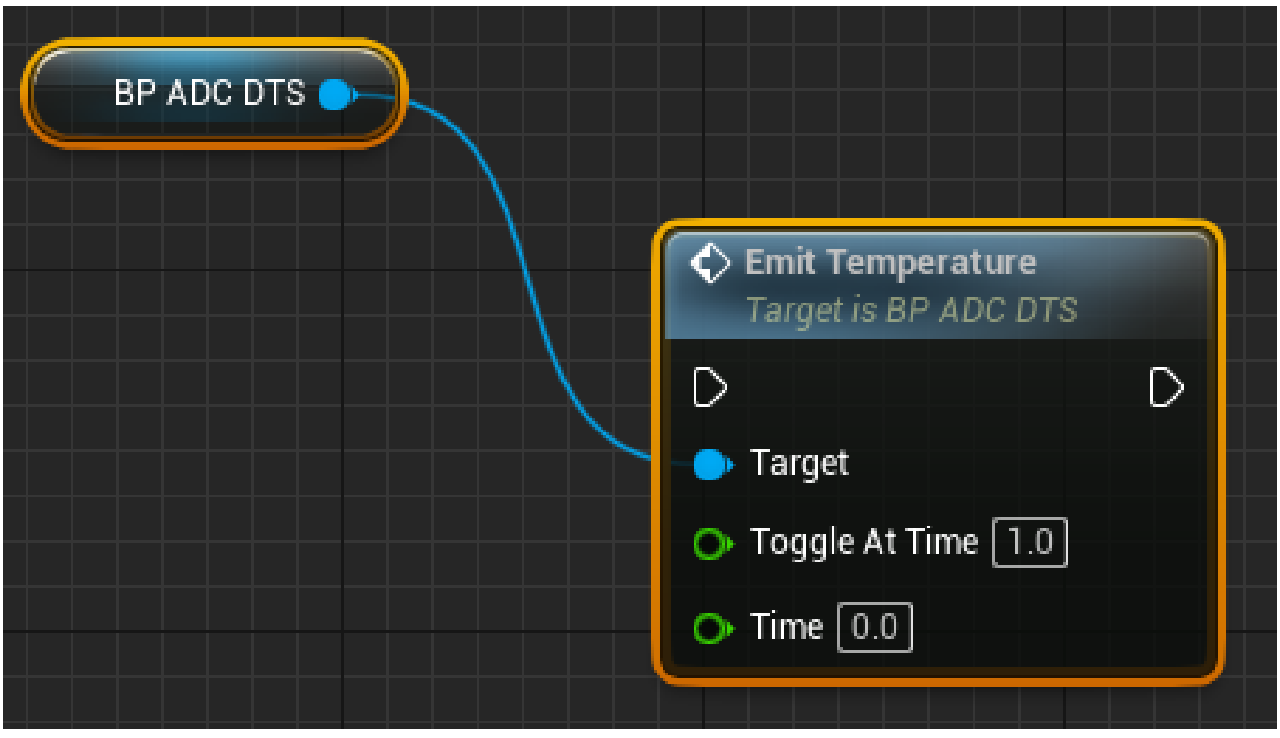
Receive Temp



TICK FOR OBJECTS EMITTING TEMPERATURE:

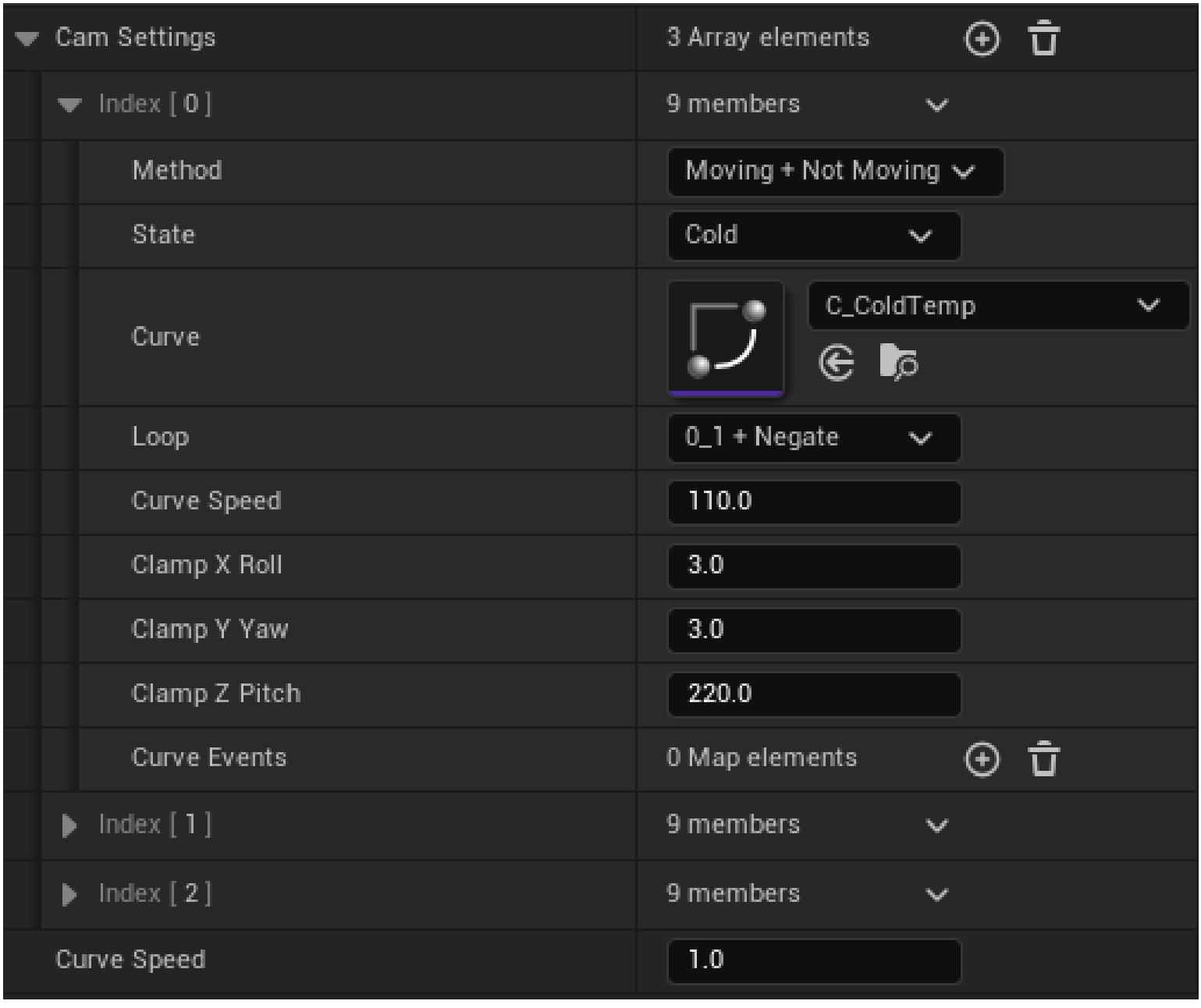
Drag **BP_ADC_DTS** into the main graph and call

Emit Temperature



Dynamic Camera System

Add the component to your character or blueprint by navigating to the + Add in the top left corner under Components tab and search for **BP_ADC_DCS** once added click on it and navigate to the details tab to find the following settings:



Method: when should the curve values be applied?
Moving, when the player is moving the camera
Not Moving, when the player is not moving the camera
Moving + Not Moving, all the time

State: Populate these states through the enum found at **ADC_Component -> Blueprints -> Enums -> E_Player_States** the states will allow you to define each array element in the **CAM SETTINGS;**

Curve: The vector curve to use for animating the rotation values.

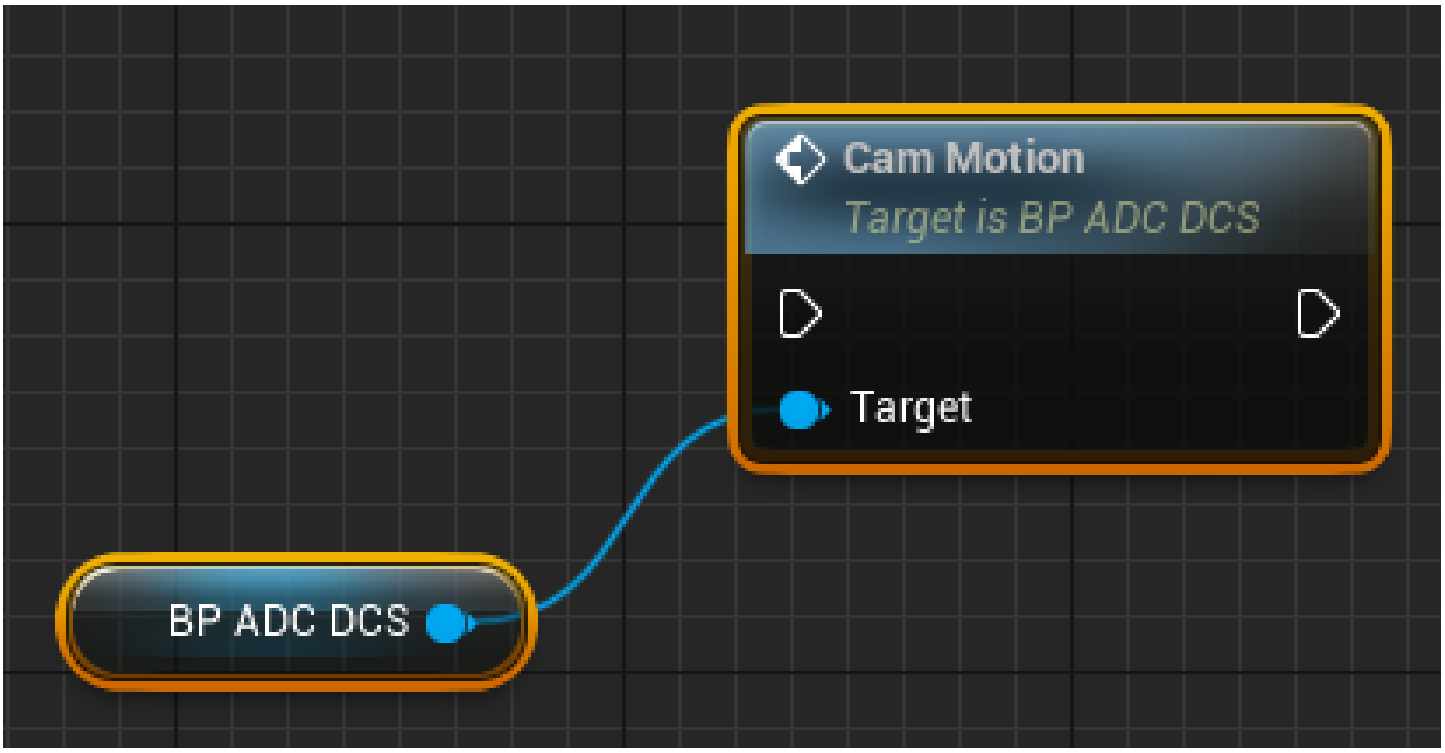
Loop: Controls the looping method used for driving the curve value.

No Loop, plays once
0_1 back and forth
0_1 + Negate, Back and negated forth (recommended)
0_0 back to back
0_0 + Negate back to negated back

Clamping: divides each axis by a certain value thus clamping it.

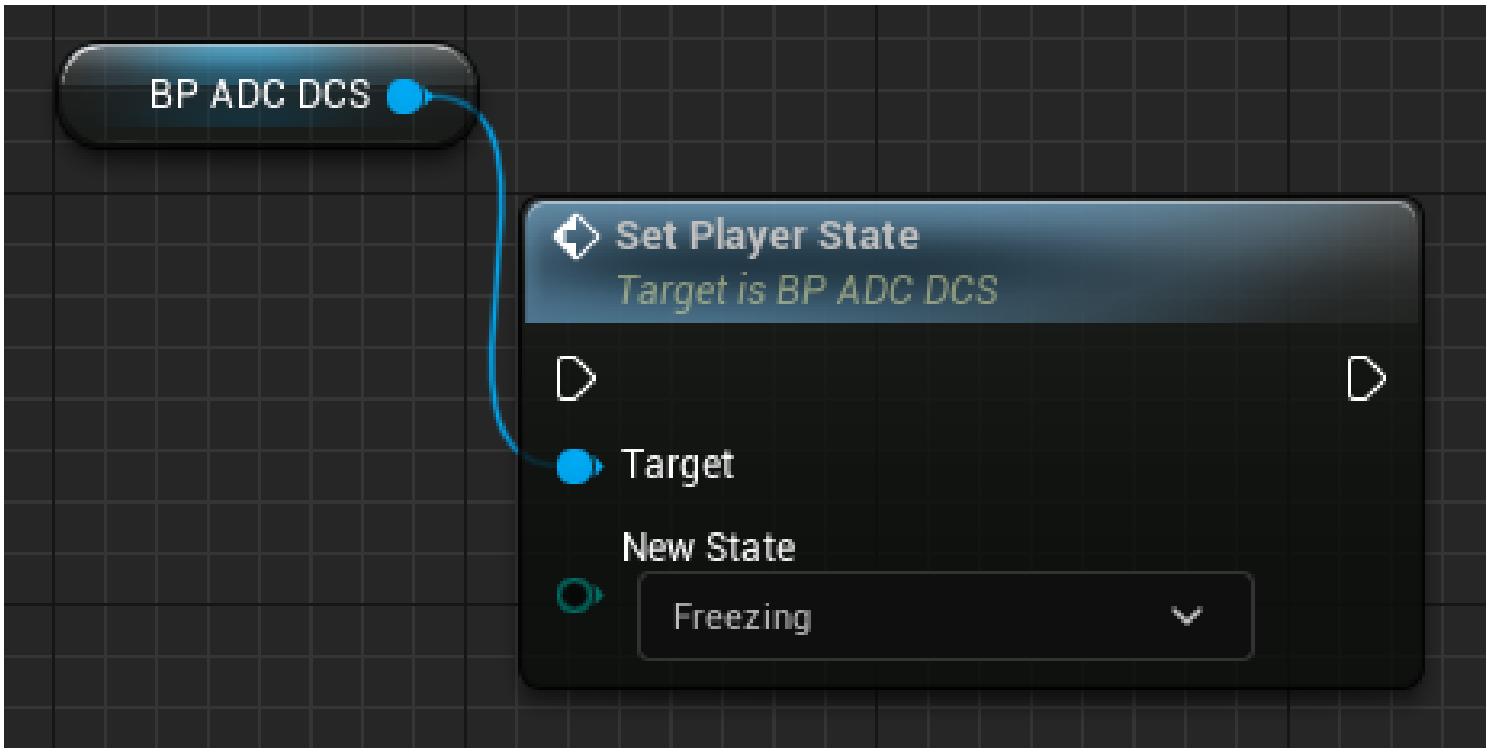
TICK UPDATE FOR CAMERA SYSTEM:

Drag **BP_ADC_DCS** into your graph and call **Cam Motion**

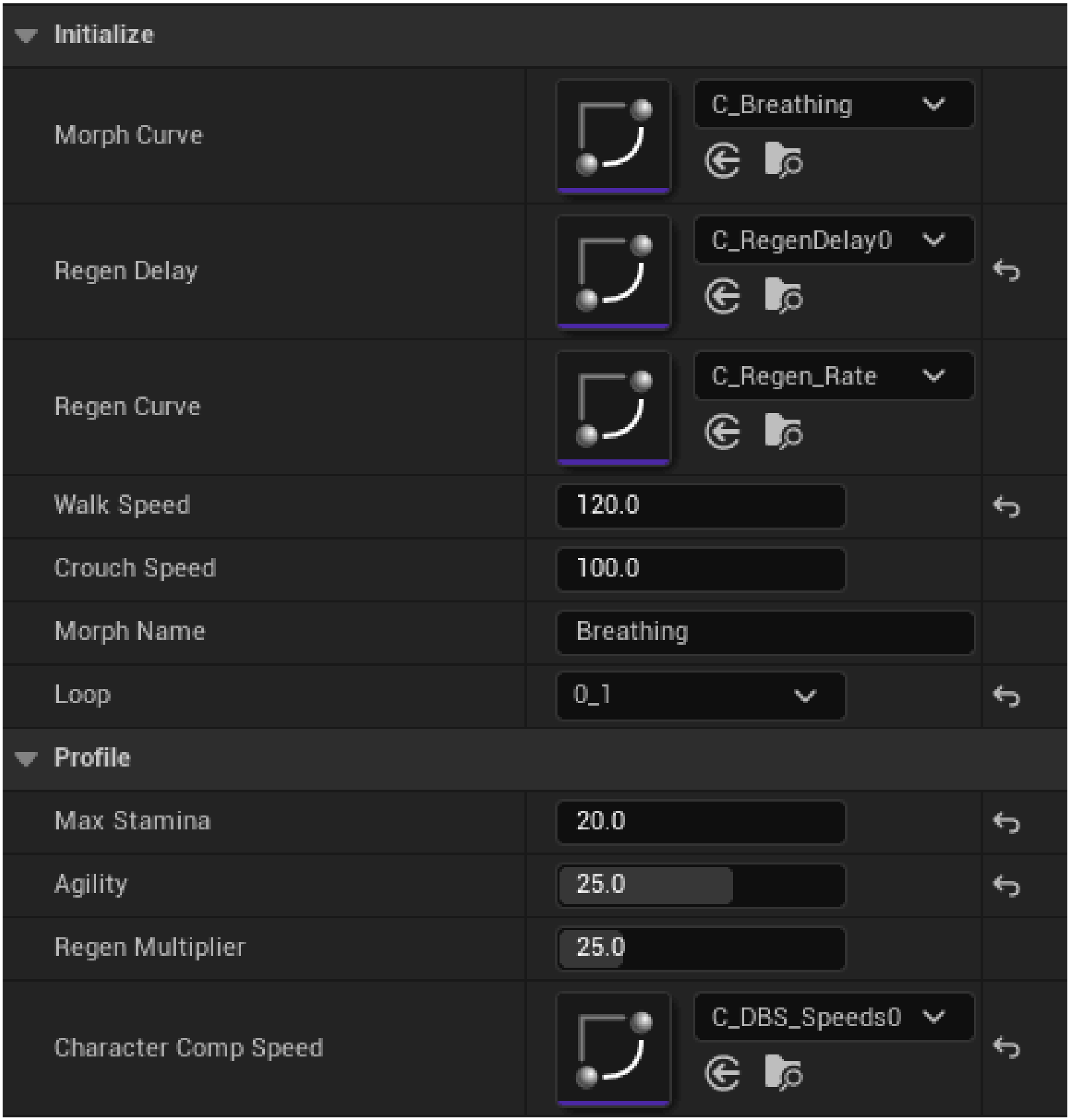


CHANGING TO A NEW STATE:

Drag **BP_ADC_DCS** into your graph and call **Set Player State**



Dynamic Breathing System



Morph Curve: curve responsible for animating the Skeletal Mesh morph target to simulate breathing

Regen Delay: this curve should be a fixed time of 0.0 to 1.0
It is responsible for delaying the stamina regeneration based on how much stamina has been burned.

Regen Curve: this curve should be a fixed time of 0.0 to 1.0
It is responsible for allowing regeneration amounts based on how much stamina has been burned.

Walk Speed: The speed point where no stamina should be burned.

Crouch Speed: The speed point where no stamina should be burned when crouched.

Morph name: The morph target used to control the breathing animation found in the skeletal mesh.

Stamina Profile

Max Stamina: The maximum allowed stamina for this character, can be edited in runtime.

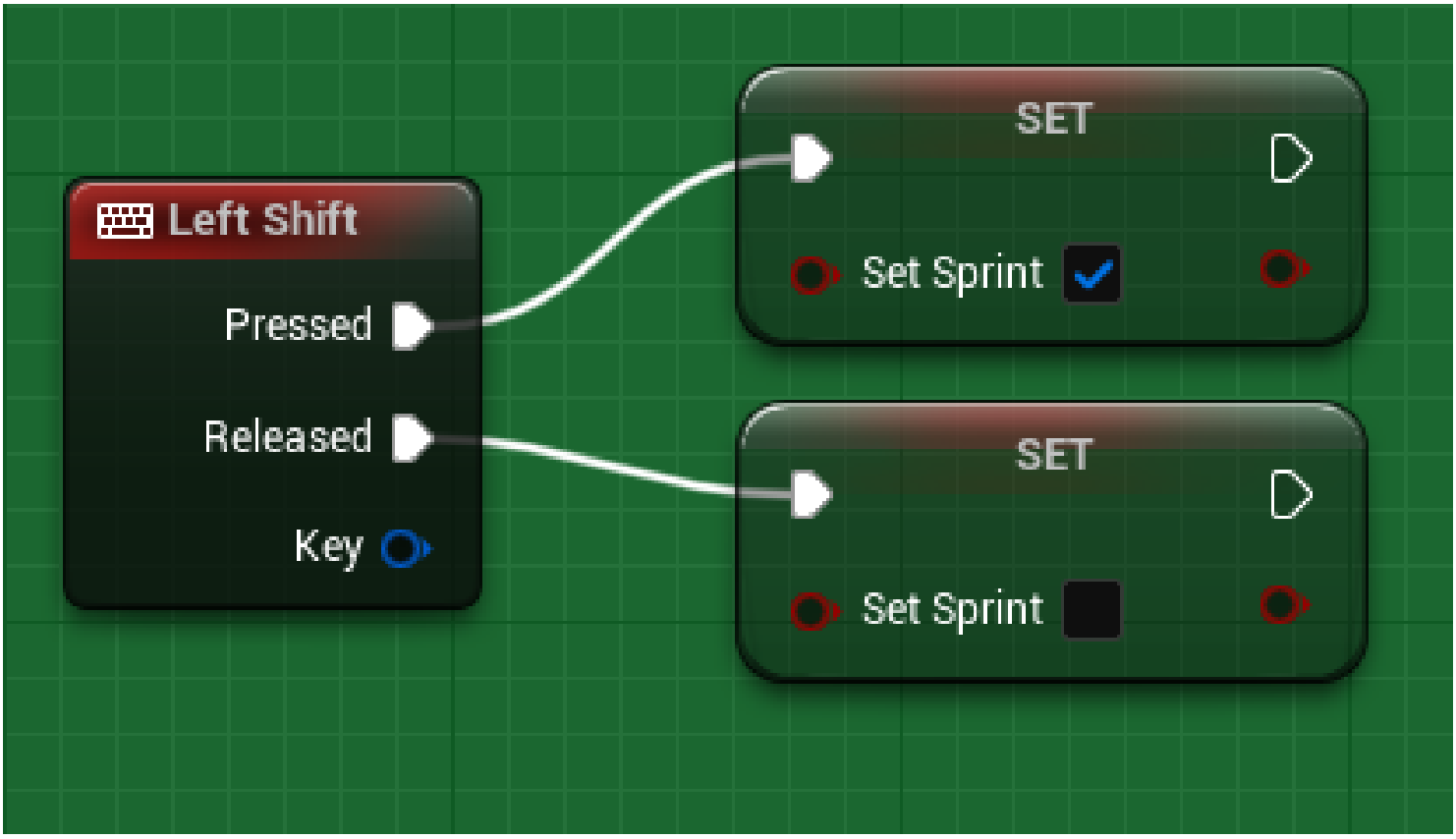
Agility: This variable controls how much stamina can be burned, the higher this value the less stamina will be burned.

Regen Multiplier: This variable controls how much stamina can be regenerated and works as an external control for managing the Regen Curve.

Character Comp Speed: This curve should be a fixed time of 0.0 to 1.0

It is responsible for allowing the maximum allowed speed based on how much stamina has been burned.

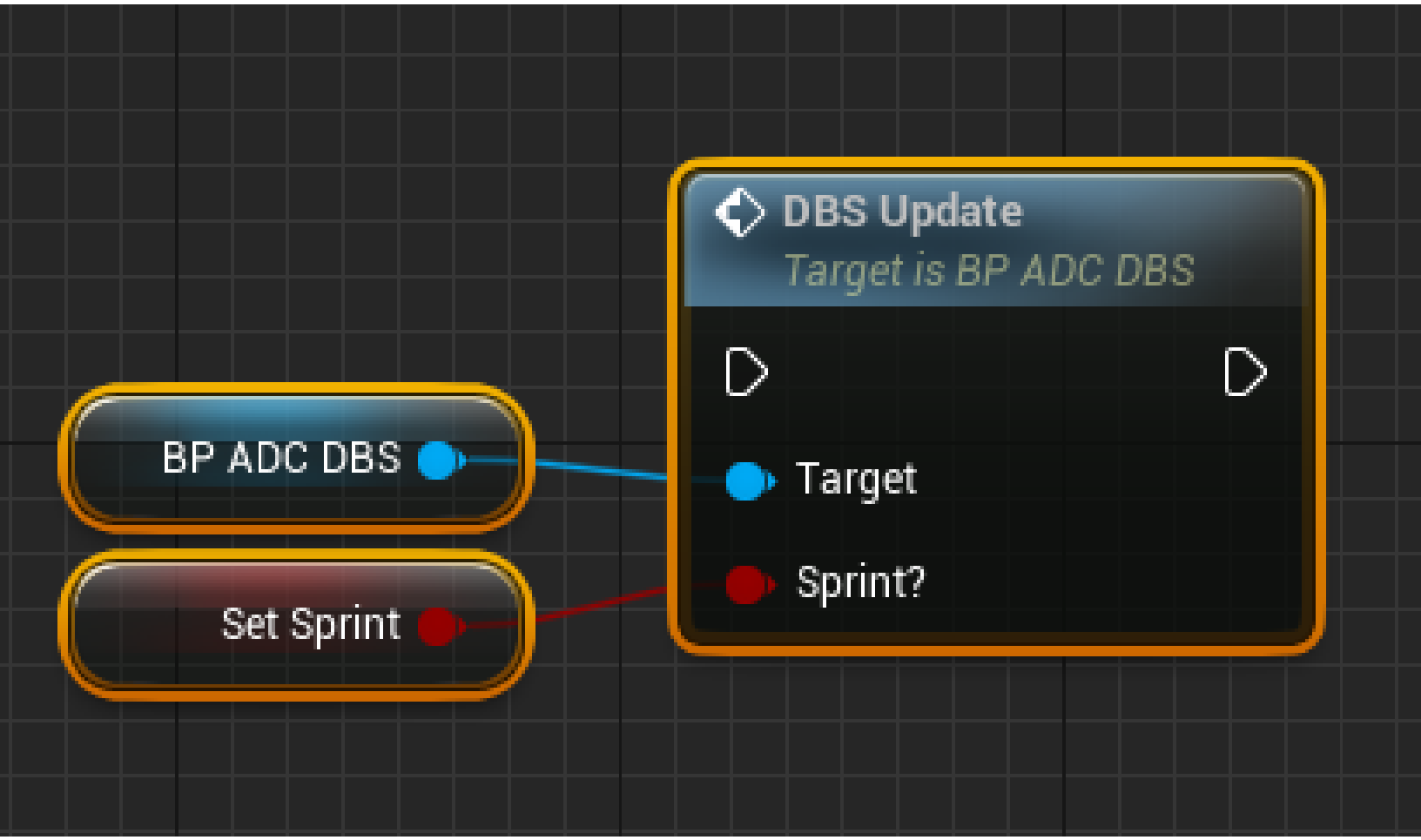
Stamina Setup



After the component has been configured through the details tab, it must be set up through the event graph along with a boolean to instruct the component when the character wants to run or walk.

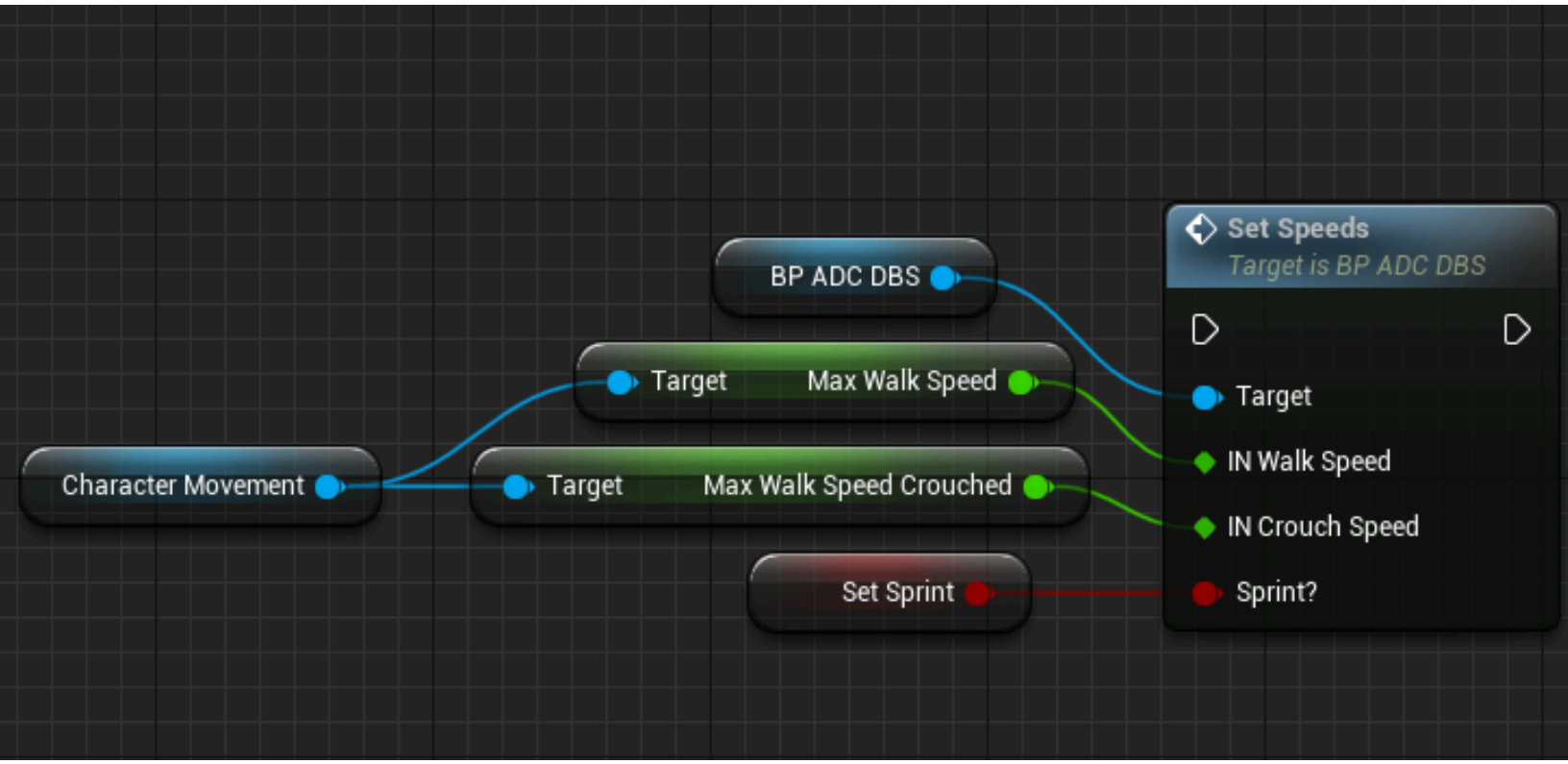
After the required boolean is set up, the component is ready to be injected into the tick function.

TICK UPDATE FOR BREATHING:



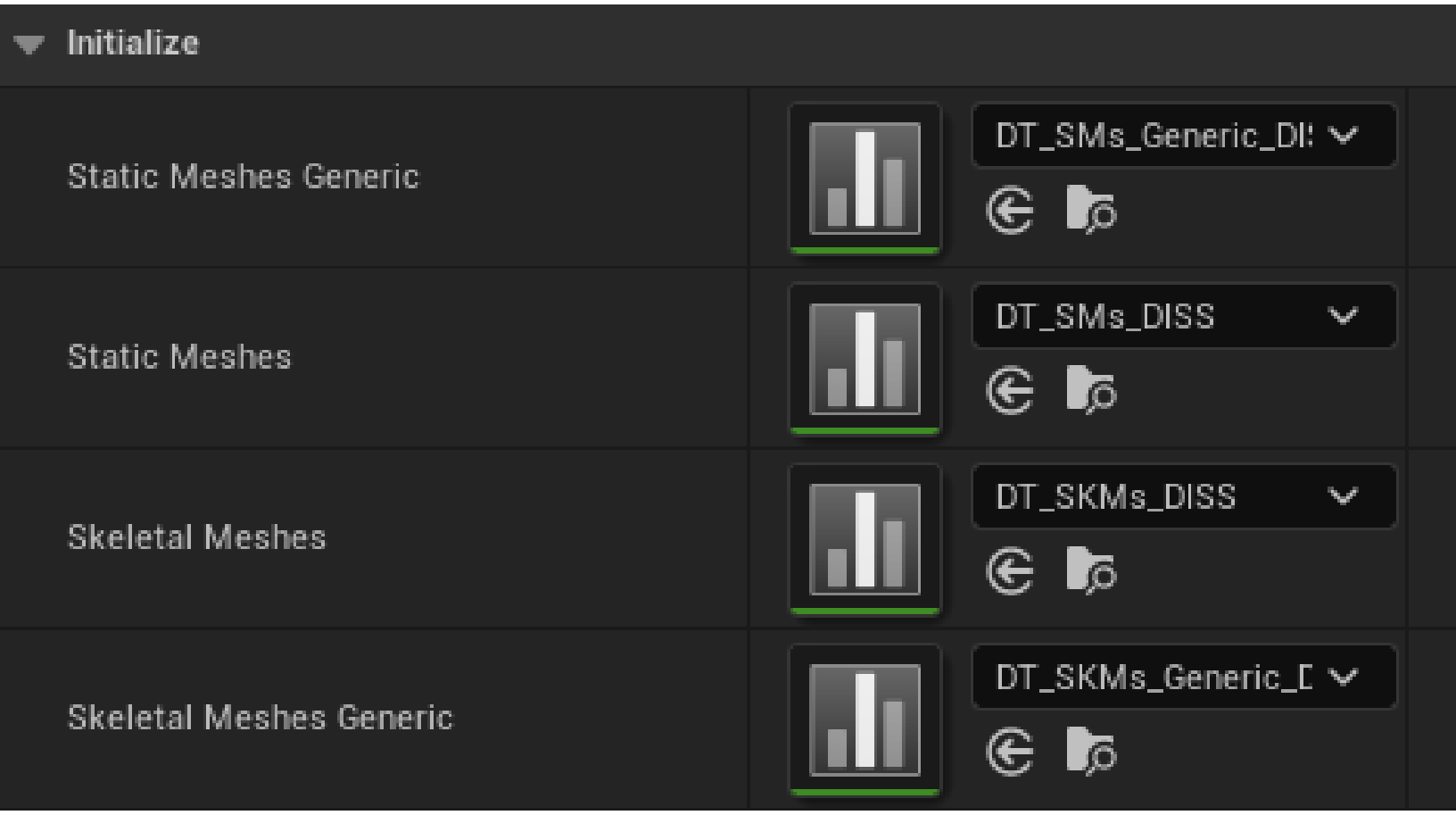
This function will update the morph target animation and current stamina using the initialization settings configured through the details tab, it can also be used to drive the fog VFX of the character example case found in BP_ThirdPersonCharacter in the Demo/DemoBPs folder

TICK UPDATE FOR CHARACTER SPEEDS:



The SET SPEEDS function will update the character speed via the Character Comp Speed Curve configured through the details tab of initialization step, it should be injected with the walk speed and crouch speed of the character movement component and instructed on when to sprint via the boolean controlled by the Character.

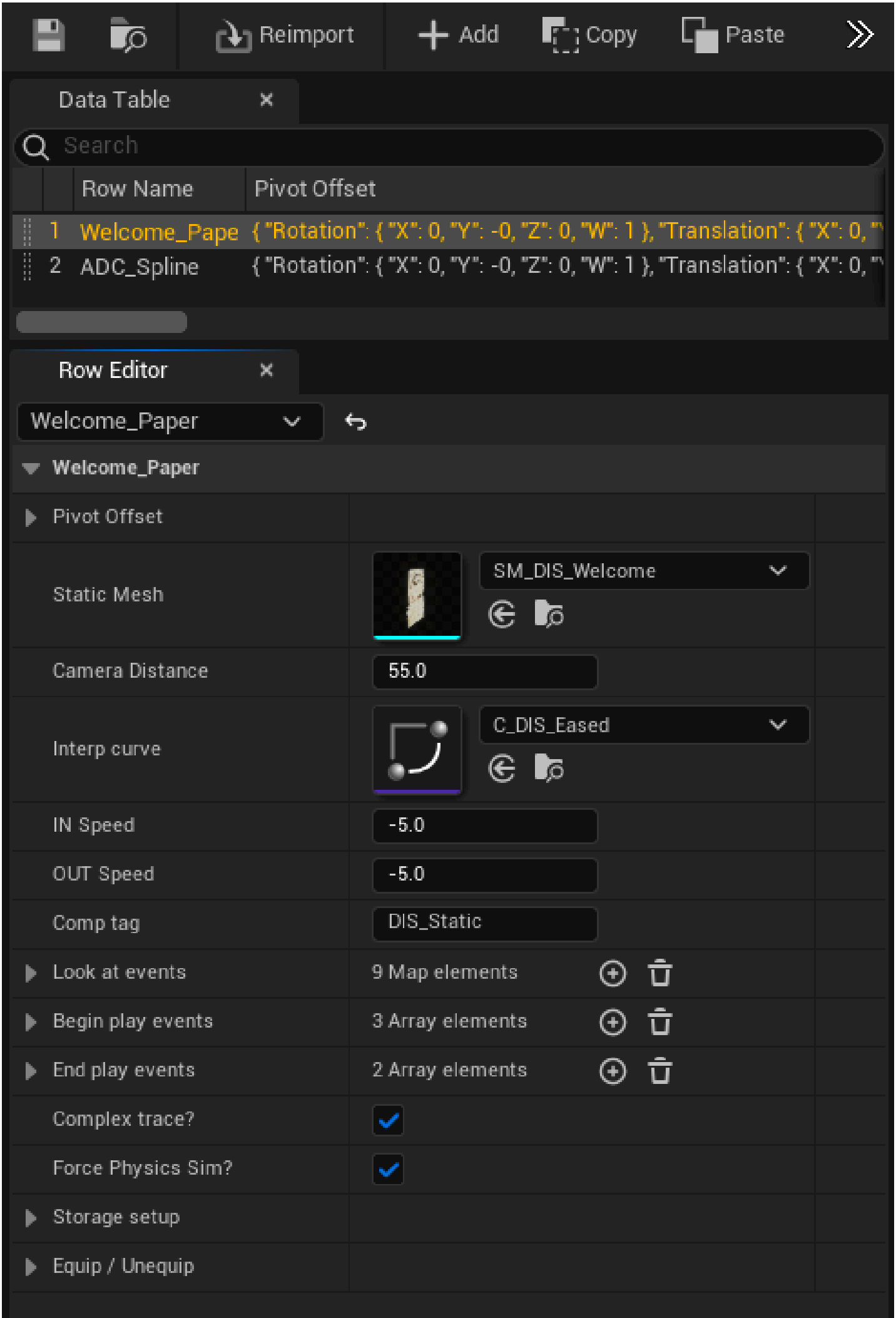
Dynamic Interaction System



The interaction system is handled through BP ADC DIS

The component must be added to the player's blueprint first and initialized through its details tab by assigning the Data Tables in the required slots.

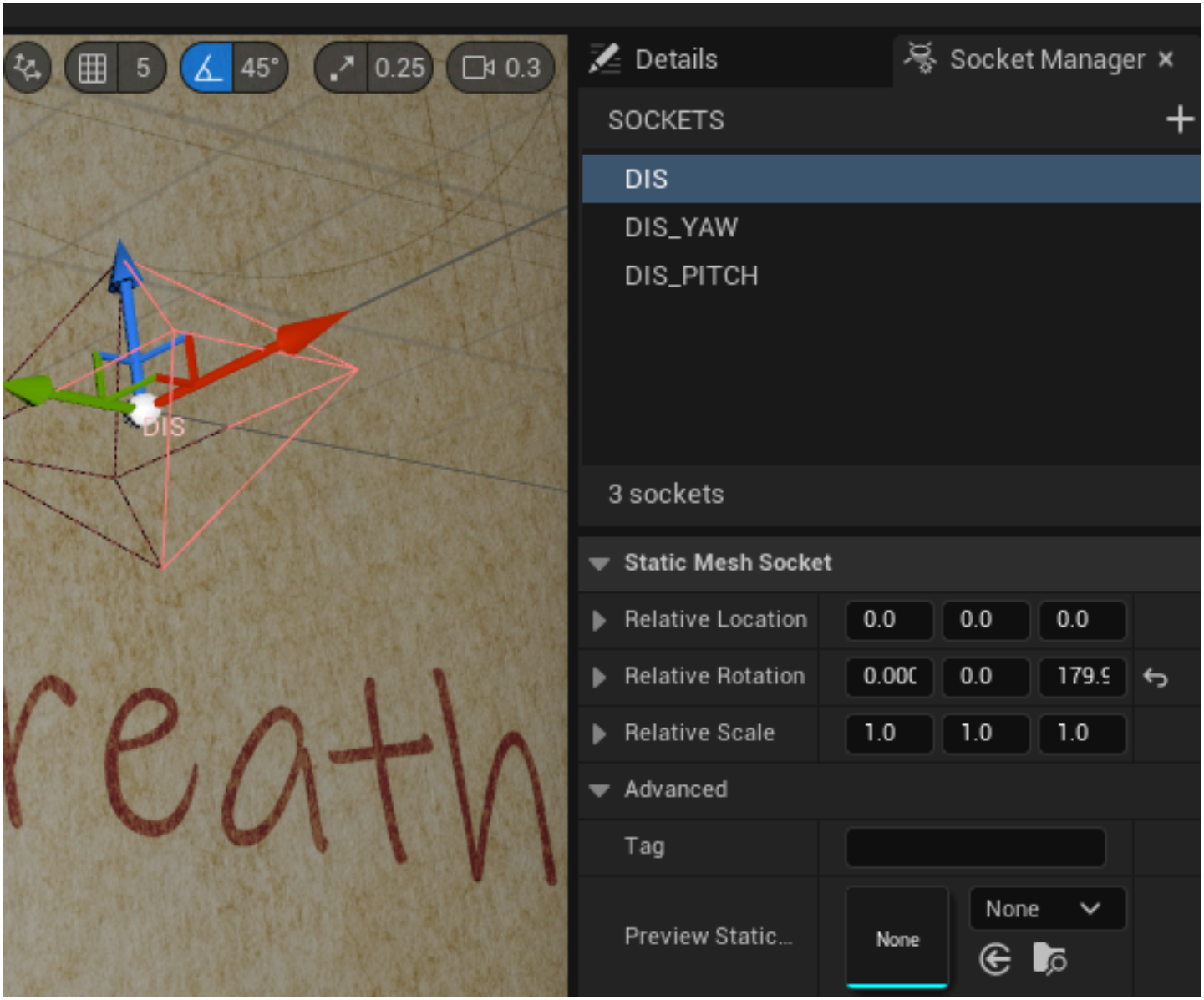
Generic vs Non-Generic: The generic Data Tables will not use tags to find settings and so it is good to use for HISMs while the non-generic Data Tables will use tags to find settings allowing special events to take place for example; objectives, scripted events


Reimport
Add
Copy
Paste
Data Table
Search
Row Name
Pivot Offset
1 Welcome_Pape { "Rotation": { "X": 0, "Y": -0, "Z": 0, "W": 1 }, "Translation": { "X": 0,
2 ADC_Spline { "Rotation": { "X": 0, "Y": -0, "Z": 0, "W": 1 }, "Translation": { "X": 0,
Row Editor
Welcome_Paper
Welcome_Paper
Pivot Offset
Static Mesh
SM_DIS_Welcome
Camera Distance
55.0
Interp curve
C_DIS_Eased
IN Speed
-5.0
OUT Speed
-5.0
Comp tag
DIS_Static
Look at events
9 Map elements
Begin play events
3 Array elements
End play events
2 Array elements
Complex trace?
Force Physics Sim?
Storage setup
Equip / Unequip

Inside the Data Table

Assigning the mesh: The mesh to be used with DIS must have the following sockets assigned to them in order to be registered correctly;

DIS: The new pivot for the mesh and the X axis camera should face.
DIS_YAW: The Z axis we should rotate around
DIS_PITCH: The X axis we should rotate around



Camera Distance: The distance away from the camera when the item is toggled for examining into camera view.

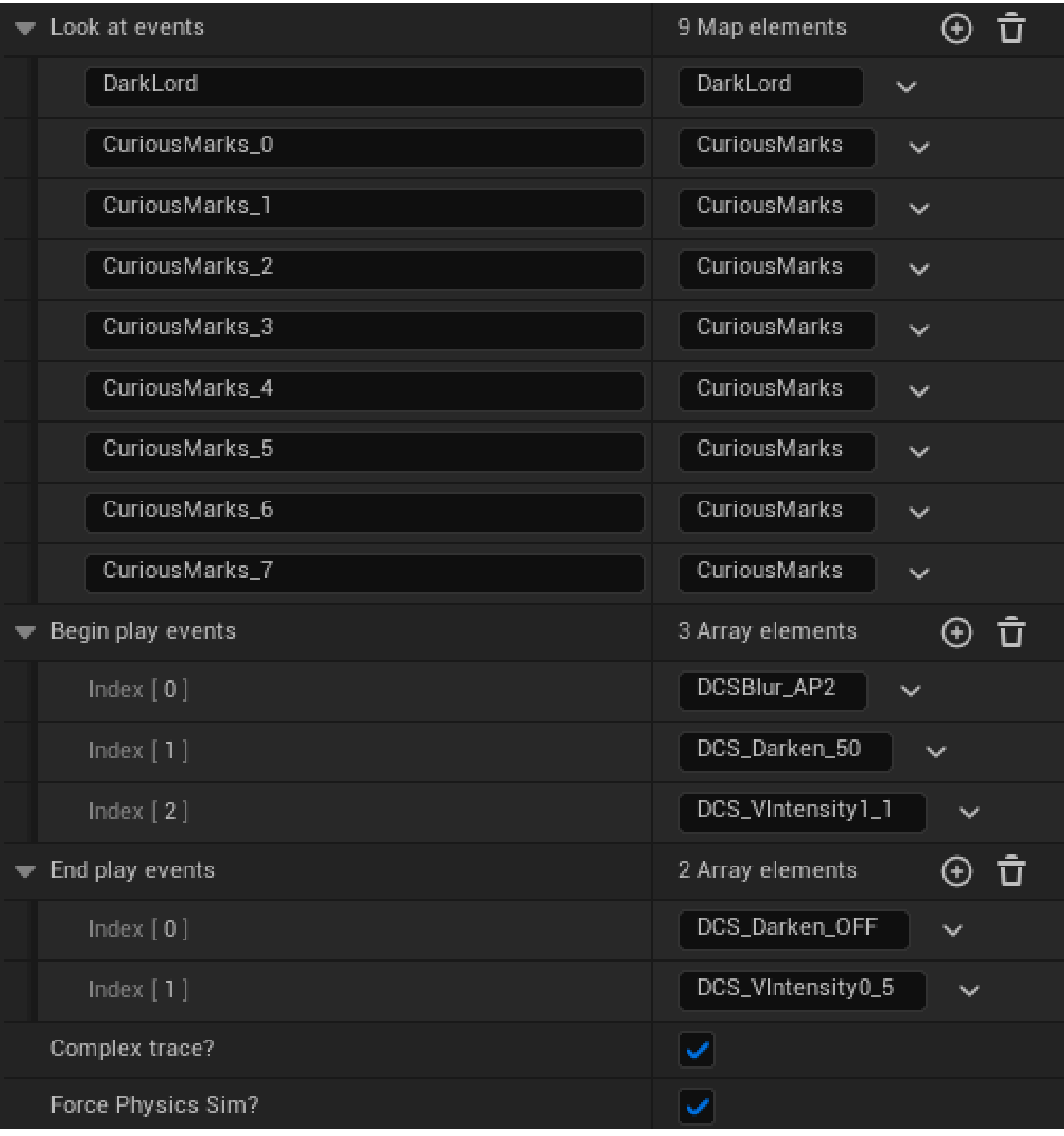
Interp Curve: The easing curve for interpolating between the rest point and the camera point to bring the mesh to.

Comp tag: This will only be used inside Non-generic Data Tables, this can be used to identify objective / quest items and / or certain events depending on the developer, tagging will never be used for HISMs.

HISMs: When interacting with HISMs make sure the ***"Multi Body Overlap"*** Box is set to TRUE on the HISM component also make sure not to move the HISM component around after it has been converted.

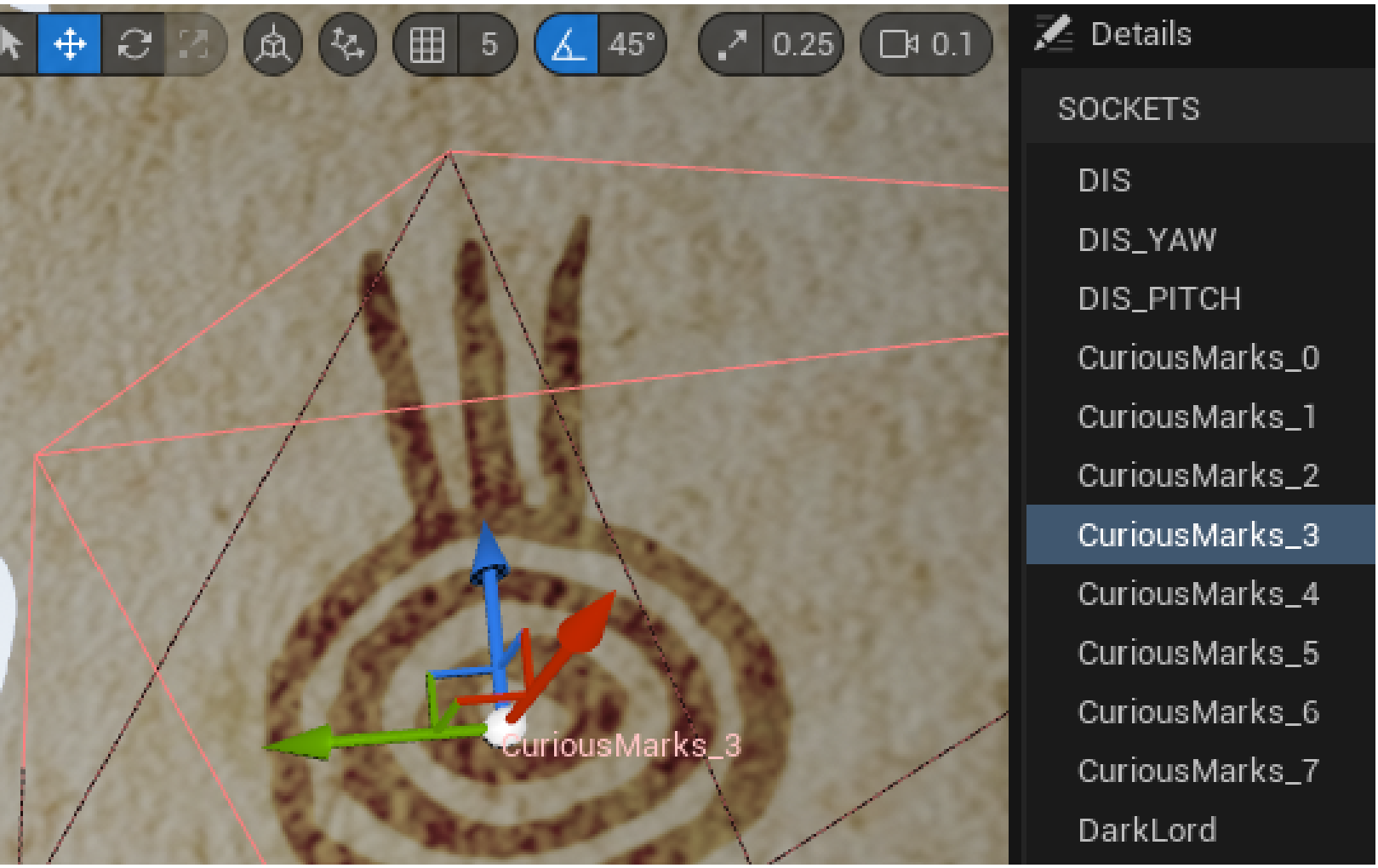
HISMs will always use the Generic Data Table for interaction.

Special Events



Look at events: to make use of this feature, a socket must be assigned to the desired location with the X axis facing the desired angle where the camera should be looking at in order for the event to fire.

After the socket is assigned, simply insert the socket name to the left and the event name to be fired.



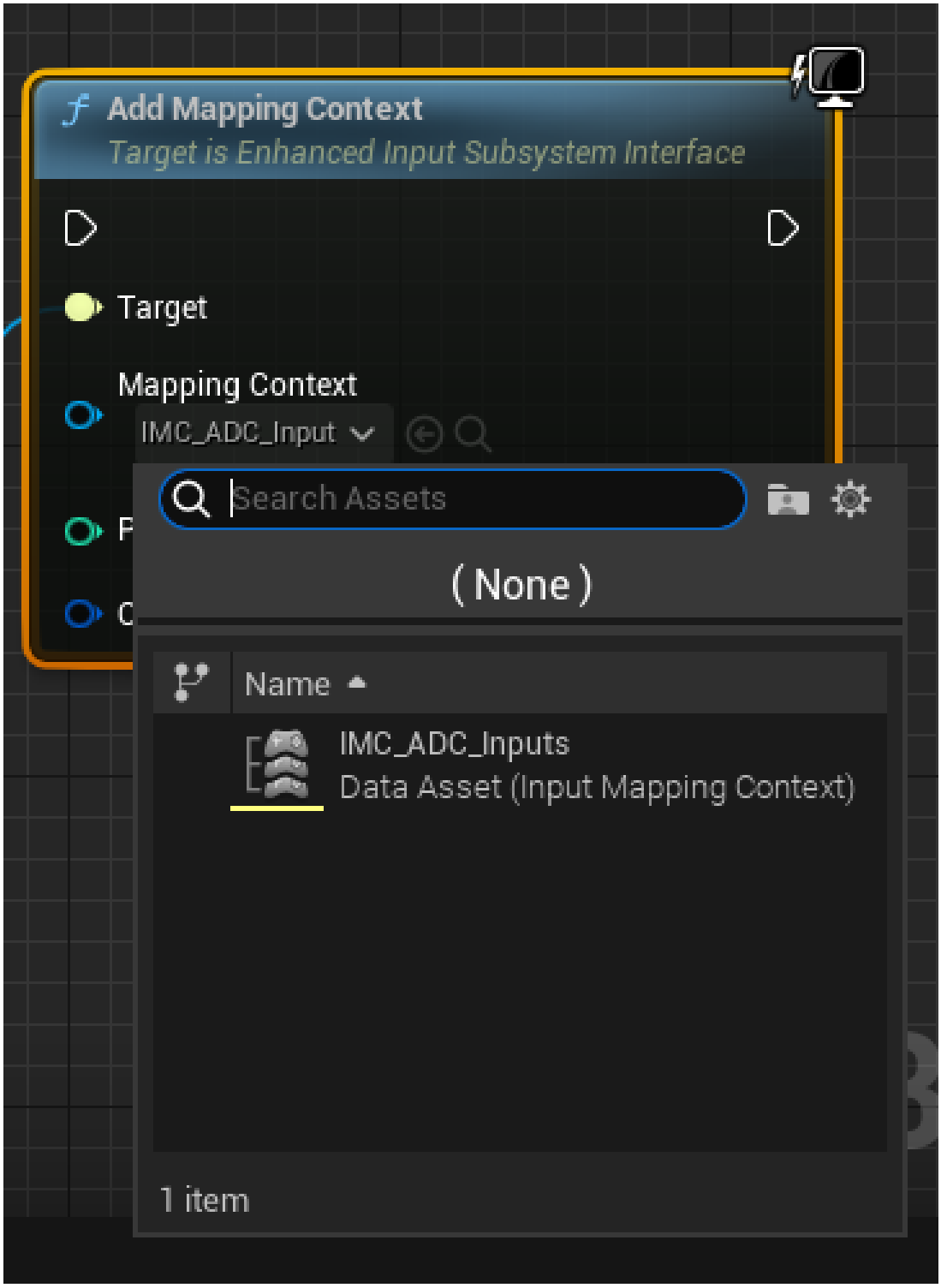
Begin Play events: These events will fire when we enter examination, Dynamic Camera Component is integrated into this to showcase the use of this system.

End Play events: These events will fire when the item is placed back or collected whilst it is being examined, these events will not fire if it is collected without examining first.

Complex trace: To be used along with post processing and for look at events in order to work properly.

For physics sim: When the item is dropped after it was being dragged around physically and it was not simulating physics by default, should we still force it to simulate physics?

The rest of the settings in the Data Table are relevant to other components and will be documented in their respective index.



To make correct use of all the interaction methods in ADC, Enhanced Input Plugin must be enabled in the project and the mapping context registered correctly on Begin Play of the Player Character

Refer to the character demo BP on how to set up the inputs correctly.

Dynamic Storage System

Dynamic Equipment System

Dynamic Doors System

Paired Animations System

Dynamic Health System

Dynamic Status Effects

Functions Library

- Events & variables overview -

BP_ADC:

Create SM DMI(s): used for auto creating dynamic materials for static meshes

Create SKM DMI(s): used for auto creating dynamic materials for skeletal meshes

Create GC DMI(s): used for auto creating dynamic materials for Chaos

Create Niagara DMI(s): used for auto creating dynamic materials for Niagara comp

Auto bool - if true find all relevant comps and convert their materials to dynamic

Specify Index ? - if true user must add numerical tags in the relevant component to convert the specified material index of said component

==========

ADC Init - used for allowing material, niagara and transform changes specified by user

Auto ADC tag? bool - if true tag all components system can manipulate and allow changes to happen on them through ADC Defaults variables and ADC Settings Variable

ADC Settings - used for allowing user customization of materials, niagara, light and animating their transforms, also used for scaling audio and animating its location / rotation

Is On? - used for toggling timeline on/off, controls time flow of Timeline Dynamic Power curve

Toggle Anim #'s - used for stopping / starting movement when desired, example of use found in **BP_MagicPlatform**

Light Flicker Settings - used for light color, temperature and intensity flickering

ADC Timeline 0 - used for driving changes on dynamic parameters using ADC Settings specified by the user.

ADC Timeline 1 - functions the same as index 0, used for mixing different values on different material parameters together.

ADC Timeline 2 - functions the same as index 0, used for mixing different values on different material parameters together.

thank you for supporting this ever growing library

Roadmap **ADCs**

! = High

| = Medium

? = Low

Spline Animate - ✅ 1.1

Sound Occlusion + MetaSounds Integration - ✅ - 1.2
Provide timeline presets to use in and outside of ADC - 1.2.4 ✅

Global Events System - 1.4.4 ✅

Dynamic Sound System - 1.5.4 ✅

Dynamic Interp - 1.5.5 ✅

Dynamic Camera System - 1.6.5 ✅

Dynamic Temperature System - 1.7.5 ✅

Extended looping methods - 1.7.6 ✅

Dynamic breathing system - 1.8.7 ✅

Dynamic interaction system - 1.9.8 ✅

Enhanced Input plugin integration + HISM and Physics Interaction + Additional Post Process Support for DCS + Improved accuracy for curve events system
- ADC 2.0✅

Dynamic Equipment + Storage + Loot System - 2.2 ✅

Dynamic door interaction + FPS Independency - 2.3.2 ✅

Paired Animations System + Dynamic Health + Status Effects + upgraded Doors System - 2.6.6 ✅

Move interp smoothing + conversion of ADCFinterp to Play Rates + UI, Doors, Item Interaction hotfixes - 2.7 ✅

==========

Upgrade Temperature System - **!** - 2.8

Dynamic Footstep System - **!** - 2.9

Niagara Fluids plugin integration - **!** - 3.0

Autonomous AI - **?** - 3.0

Dynamic eyes system - **?**

Dynamic Morph targets - **?**

Never trust an elf!