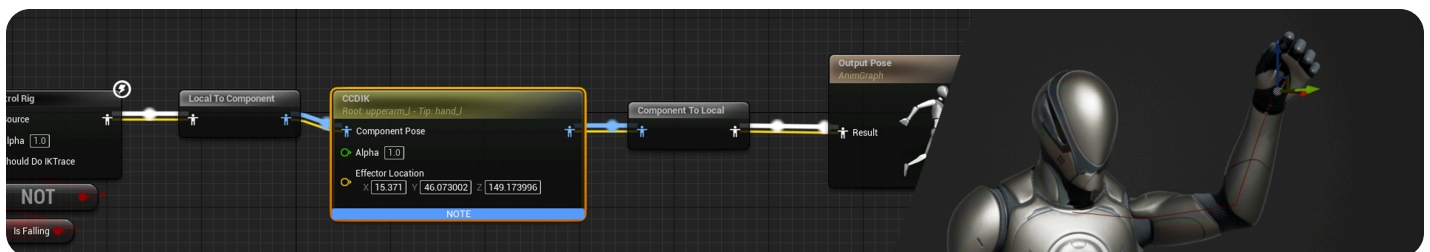


Developer
/ Documentation
/ Unreal Engine ▾
/ Unreal Engine 5.4 Documentation
/ Animating Characters and Objects
/ Skeletal Mesh Animation System
/ Animation Blueprints
/ Animation Node Reference
/ Skeletal Controls
/ CCDIK

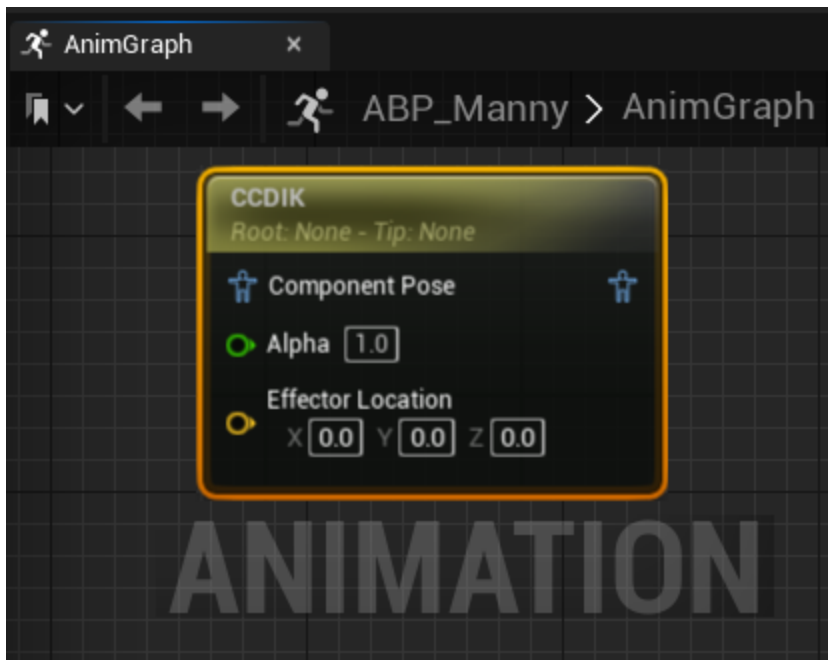
CCDIK

Describes how to access and use the CCDIK Skeletal Control node to set up and control IK chains.

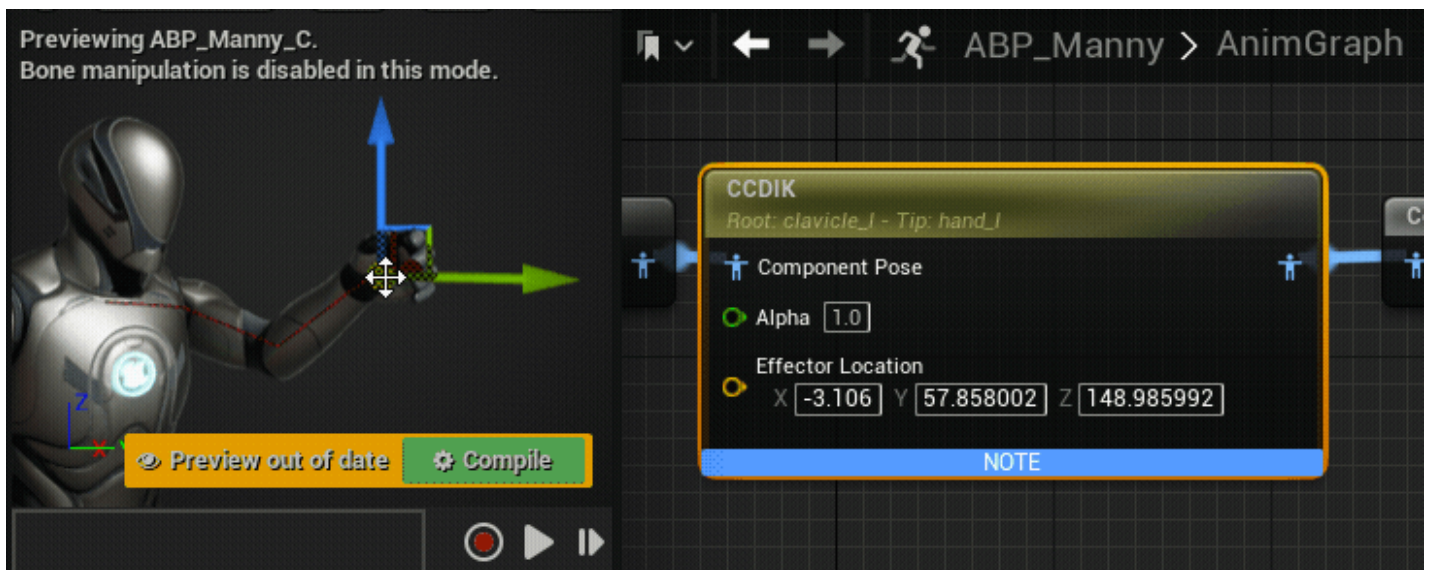


! Learn to use this **Experimental** feature, but use caution when shipping with it.

The CCDIK (Cyclic Coordinate Descent Inverse Kinematics) [Animation Blueprint](#) node is a light-weight IK solver, that you can use to drive a chain of bones to an **Effector Location**. The CCDIK node is similar to the [FABRIK](#) node, but the CCDIK provides the ability to define angular constraints which can be useful if you want to limit rotations of any affected bones.



Here the CCDIK node is selected in the **AnimGraph** and the **Effector Location** is being manipulated with the debug movement tool in the viewport window.



As the **Effector Location** is changed the selected bone chain, highlighted with the **red** debug line, moves to match the **Tip Bone** to the **Effector Location**. By using a dynamic vector variable for the **Effector Location**, and applying rotation constraints, this node can be used to make dynamic adjustments to the [Skeletal Mesh](#) to ground the character's interactions with objects in the game world.

With the **Alpha** value or pin, you can control the degree of the solved motion on the generated output pose. A value of **1** will fully use the generated output pose, while a value of **0** will output the source pose.

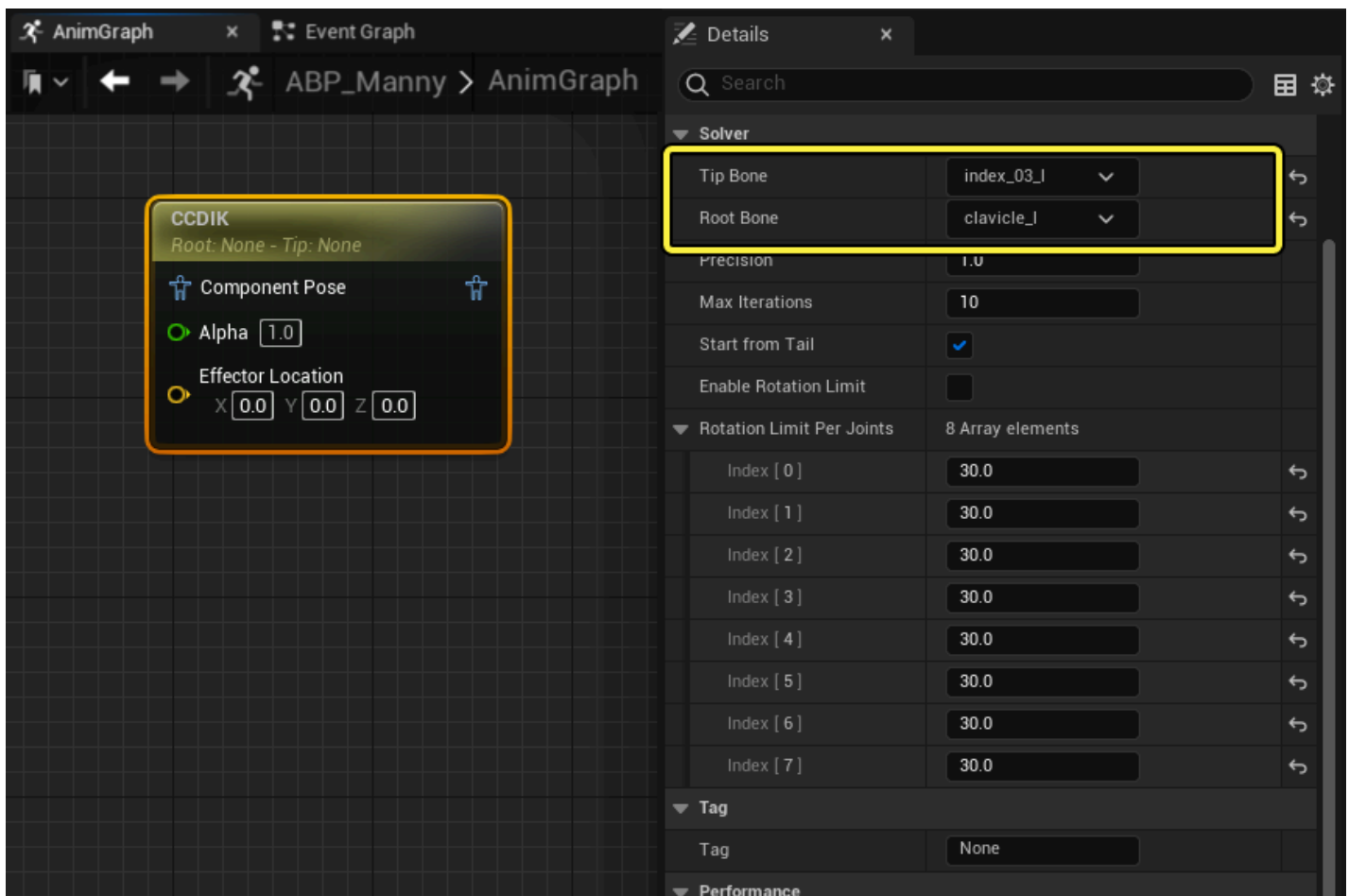
Set up

In this example workflow, the CCDIK node will be used to dynamically adjust the characters finger to press a button in the game world.

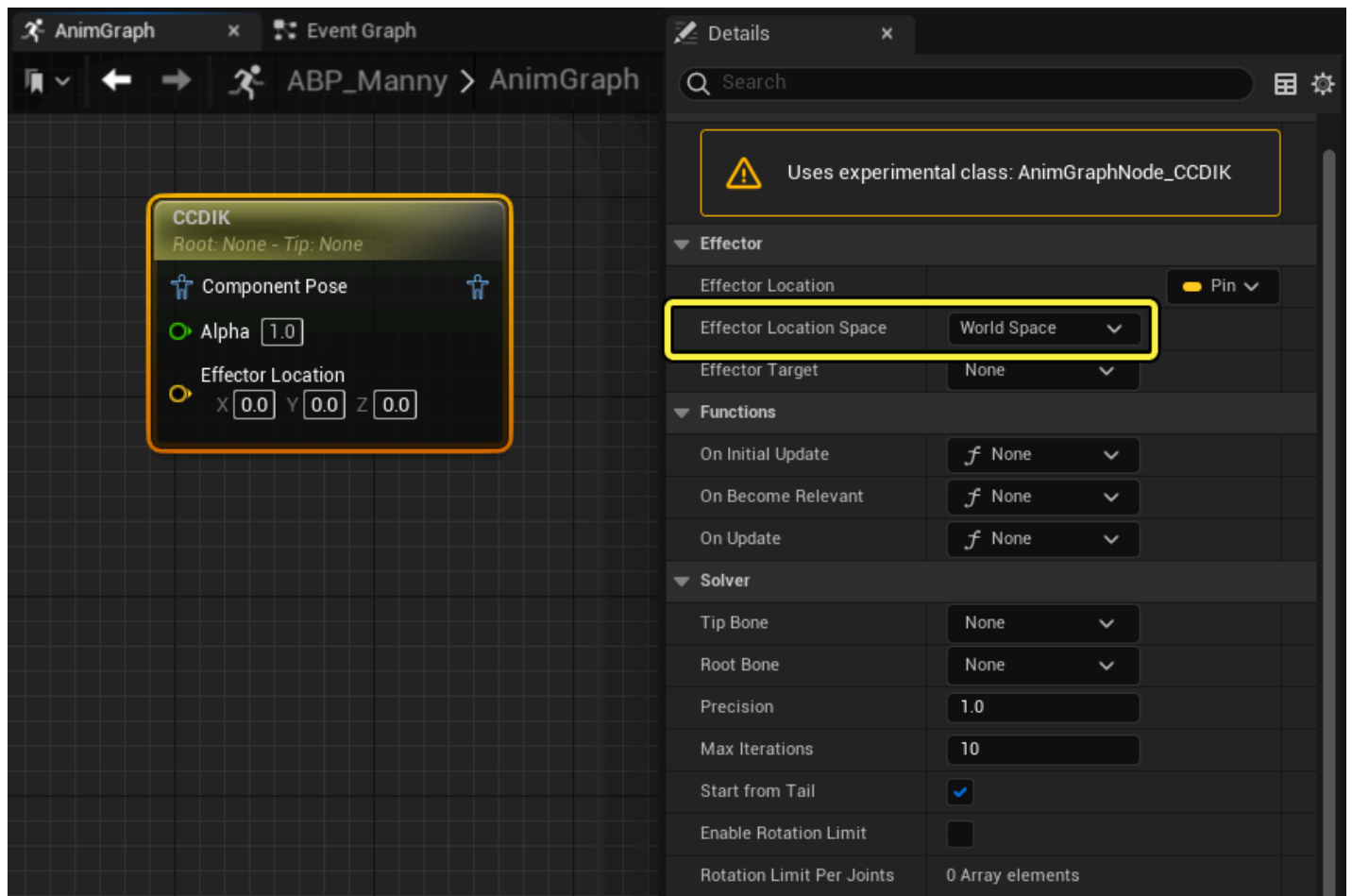
The CCDIK node operates within **Component Space**, so a [space conversion](#) will need to occur to implement the node within your character's Animation Blueprint.

After adding the CCDIK node to your [Animation Blueprint](#), select the node in the **AnimGraph** to open the **Details** panel.

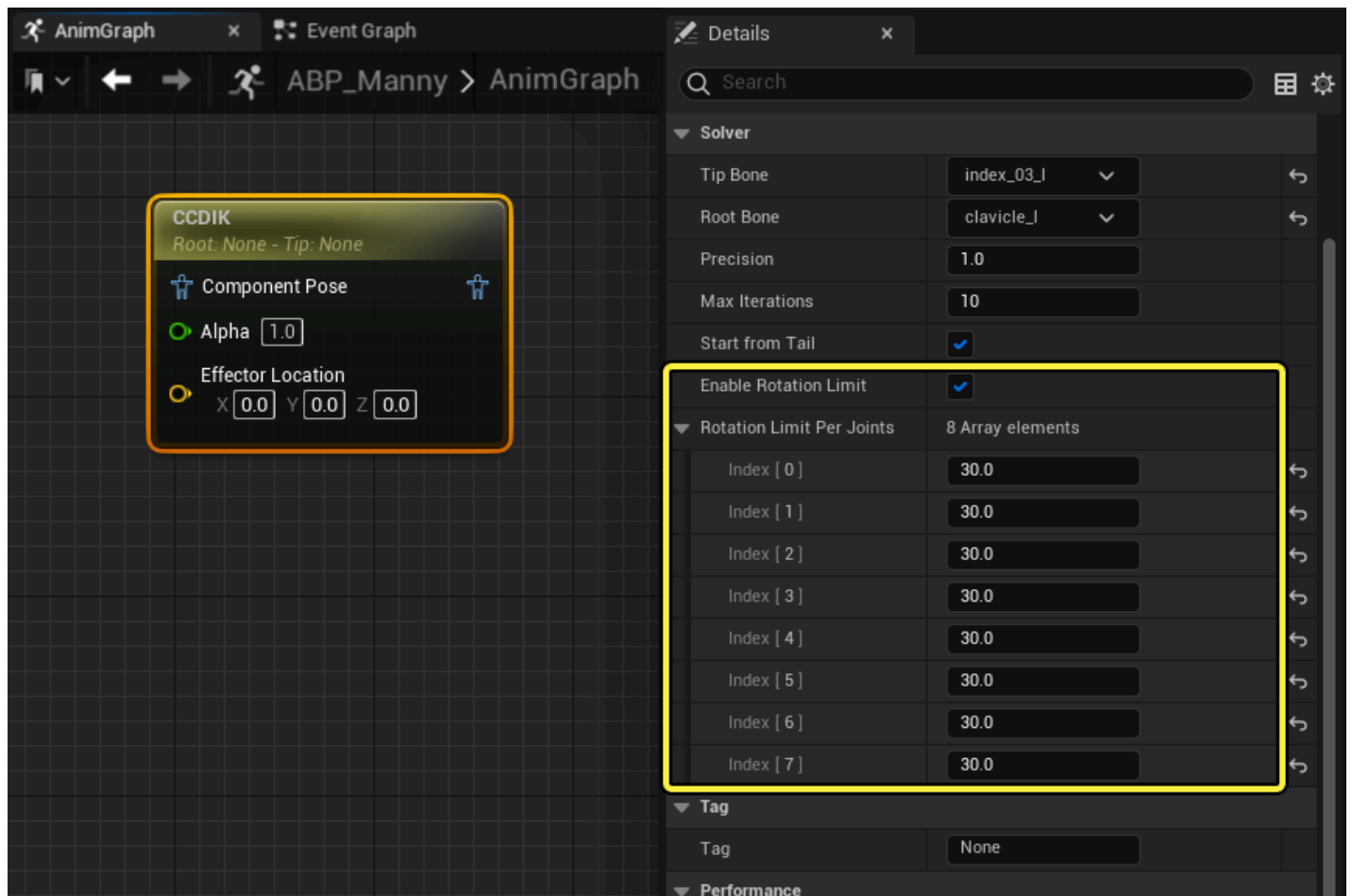
In the **Details** panel you can select the **Tip Bone**, which will be the bone at the end of your chain, and the **Root Bone**, which will ground the adjustments to a **local** root bone. In this example, the **Tip Bone** is set to the character's left index finger (`index_03_1`), and the **Root Bone** is set to the character's left clavicle (`clavical_1`).



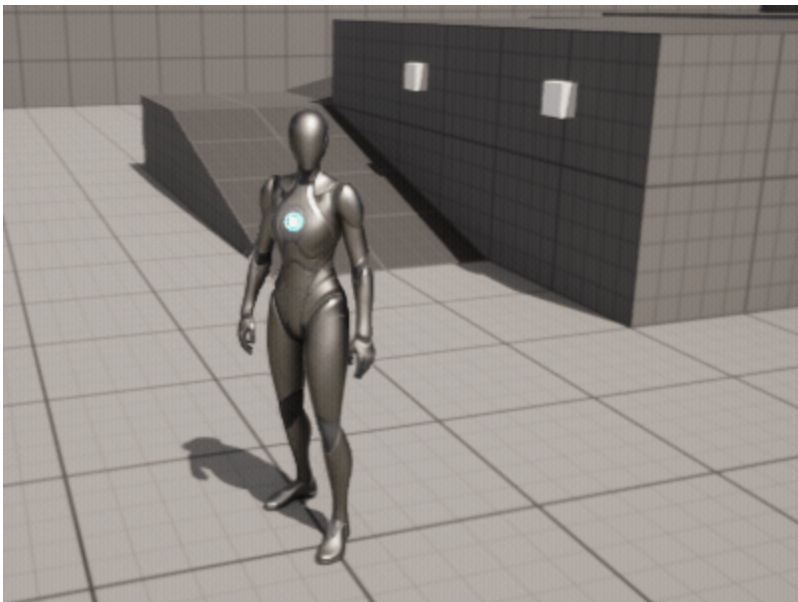
Set the **Effector Location Space** to **World Space**, to calculate the **Effector Location** vector variable in the level.



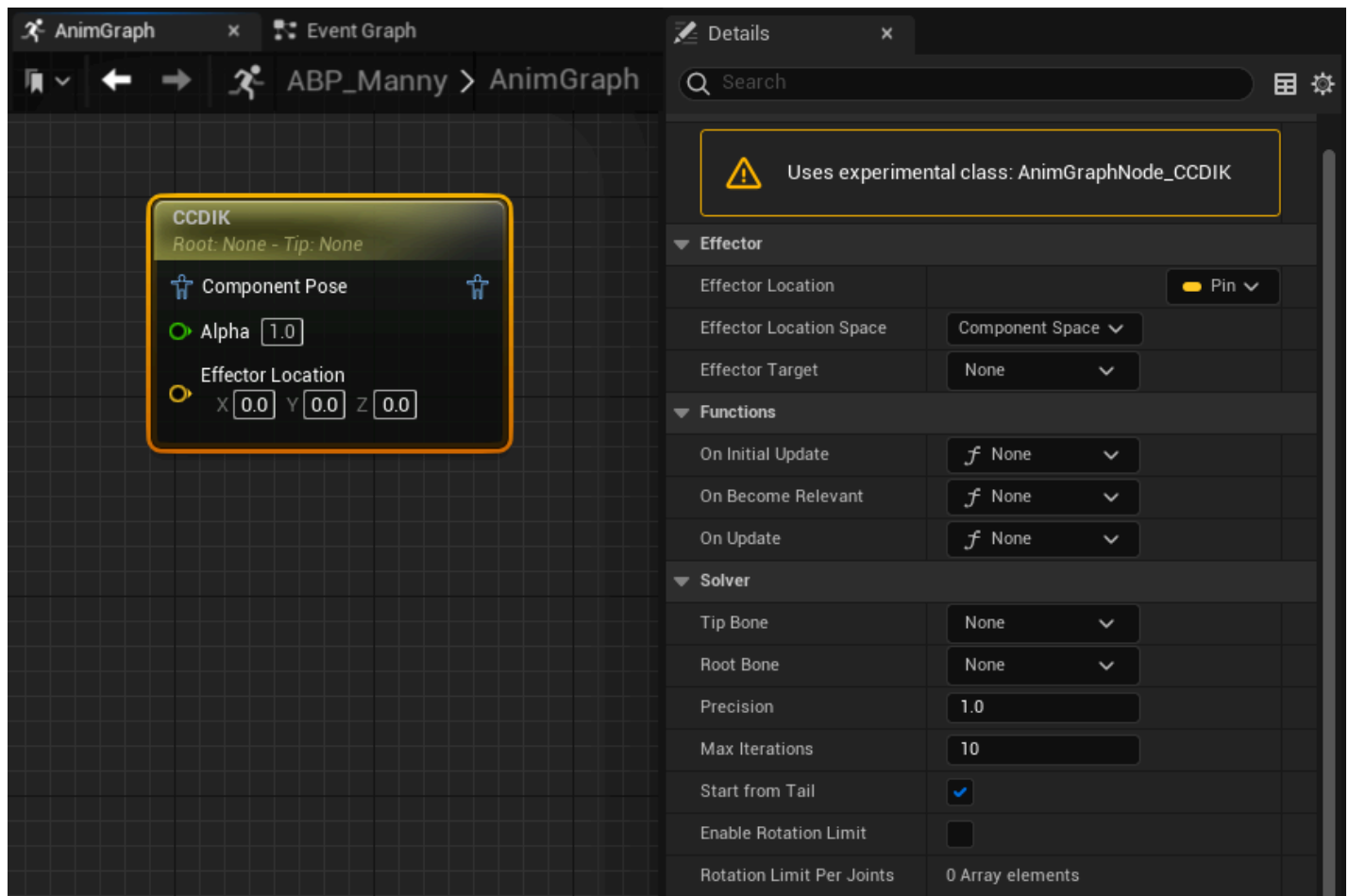
You can enable rotation limits by toggling the **Enable Rotation Limit** property and make adjustments in the **Rotation Limit Per Joints** property section. An **Index** will populate for each joint present in the [skeleton](#) between the local **Root Bone** and the **Tip Bone**. You can make adjustments to these rotations to prevent unwanted geometry overlapping and over-extension of your characters joints.



The CCDIK node will now adjust the **Tip Bone**, under the constraints of the **Root Bone** and rotation constraints, to match the location of the **Effector Location**, resulting in the character being able to dynamically interact with the button from any angle or height.



Property Reference



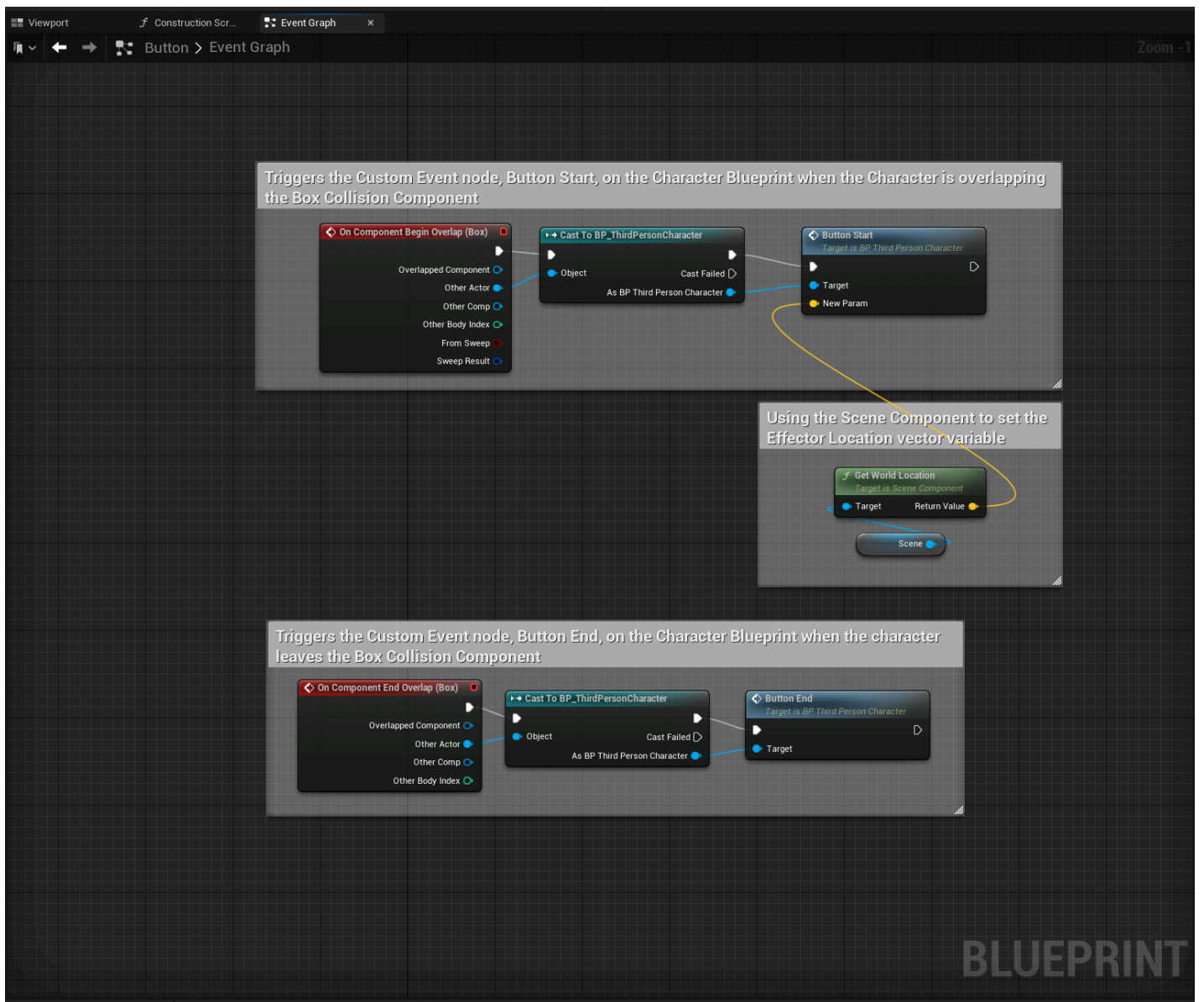
Here you can reference the CCDIK node's properties.

Property	Description
Effector Location	By default this property appears as a pin on the node in the AnimGraph . This property's value is the target point to adjust the selected Tip Bone to when solving.
Effector Location Space	Space frame of reference used by the Effector Location .
Effector Target	When the Effector Location Space is set to Bone , you can define the bone to target in this property.
Tip Bone	Select the bone from the Skeletal Mesh to adjust to meet the Effector Location .
Root Bone	Select the bone from the Skeletal Mesh to end the chain, locking the chain's motion to the local root point.

Property	Description
Precision	Here you can set the tolerance level for the Tip Bone adjustment to match the Effector Location . A value of 1 is the default value and is most commonly the best for your project. You can increase this value for more precise adjustment or reduce it to decrease the precision.
Max Iterations	Here you can set the maximum number of solving iterations the node can perform. A higher value will decrease performance but allow for more precise values to be calculated, a lower value will increase your project's performance but not allow for as much precision. A default value of 10 is suitable for more projects.
Start from Tail	When enabled the debug drawing of the chain will begin from the Tip Bone and end at the Root Bone .
Enable Rotation Limit	When enabled, the node will limit the rotation of each joint in the chain based on the Index values, in the Rotation Limit Per Joints property.
Rotation Limit Per Joints	Here you can define the rotation limits for each joint between the Tip Bone and the Root Bone in the chain. The starting Index will apply to the joint directly connected to the Tip Bone and sequentially work up the chain until the joint that is adjacent to the Root Bone .

Button Blueprint Reference

Here you can reference the button blueprint, the character blueprint and the character's animation blueprint used to create the button demonstration.



Button Actor Blueprint | Character Blueprint | Character Animation Blueprint