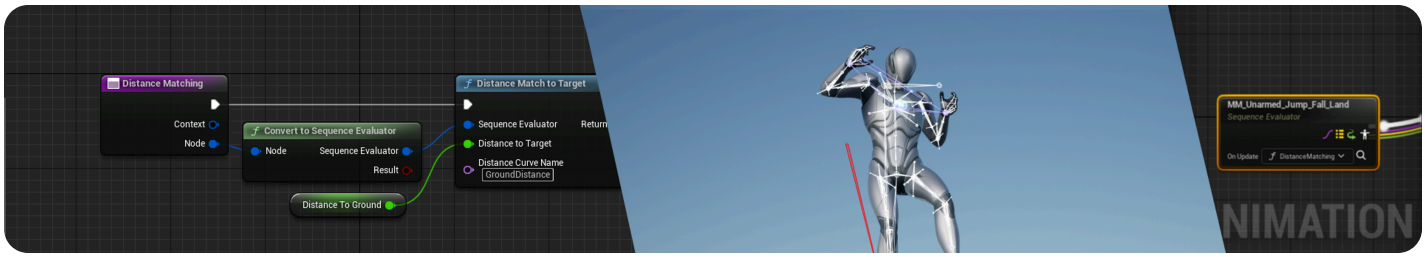
Distance Matching

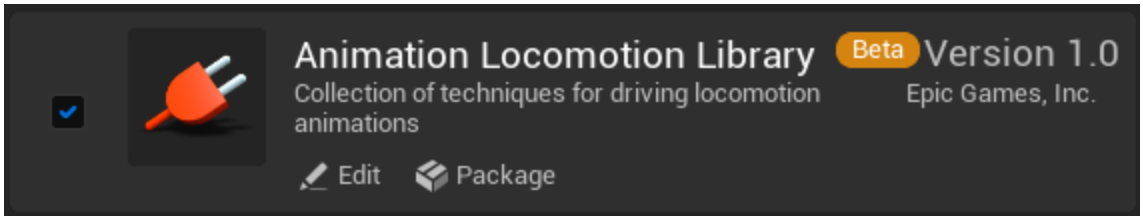
An in depth look at Distance Matching in Unreal Engine with an example workflow implementation.



With **Distance Matching**, Animation Sequences can be driven by a calculated distance variable rather than time-based linear playback. This document provides an overview of Distance Matching, and a workflow example demonstrating the implementation process.

Prerequisites

- Enable the **Animation Locomotion Library** Plugin. Navigate in the **Menu Bar** to **Edit > Plugins** and locate the **Animation Locomon Library** in the **Animation** section, or use the **Search Bar**. Enable the Plugin and restart the editor.



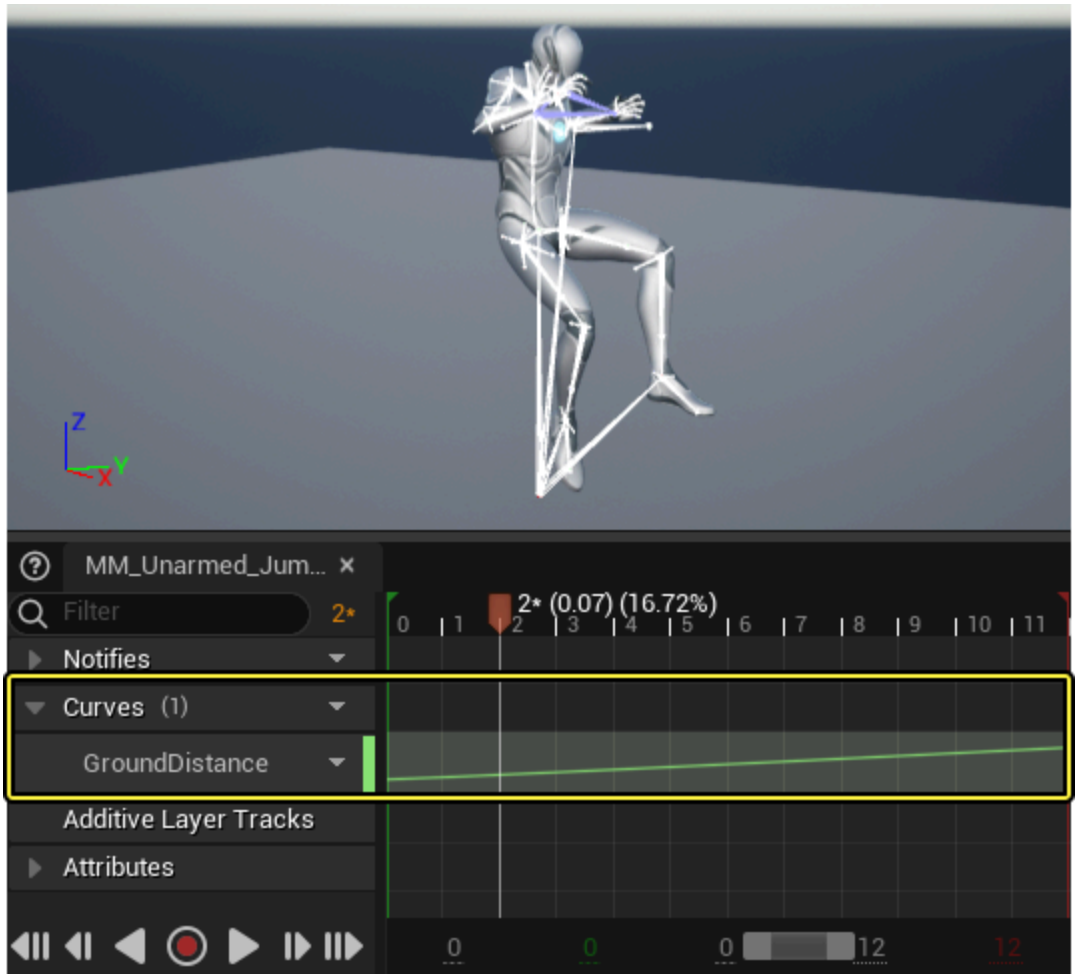
- Have an Animation Sequence.
- Have a float variable solving for the distance to or from a point.

- Distance Matching relies on Animation Blueprint and Animation Node Function, so a foundational knowledge in both is required.

Overview

With Distance Matching, you can drive Animation Sequences with the character's distance to or from a target. An Animation Sequence Curve is used within the Animation Blueprint logic, to graph and select animation keyframes based on the desired pose at specific distance values rather than time displacement.

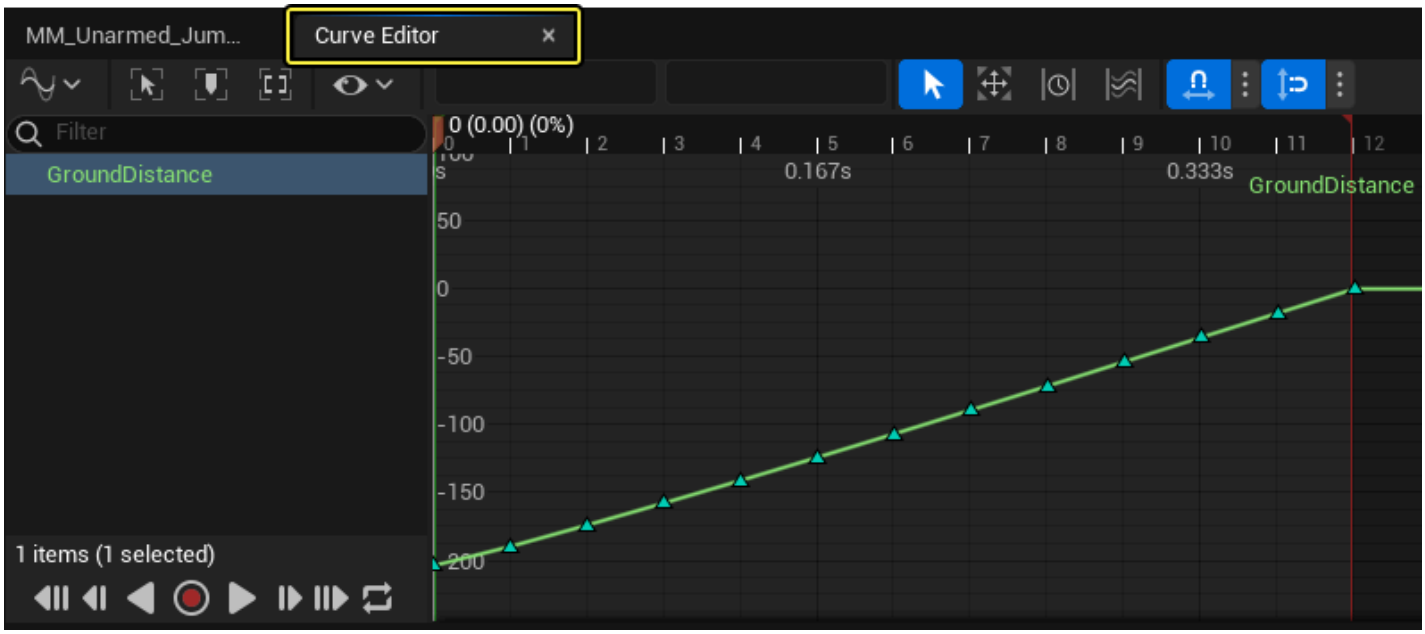
Below is an example of an Animation Sequence Editor with a curve highlighted in the Timeline.



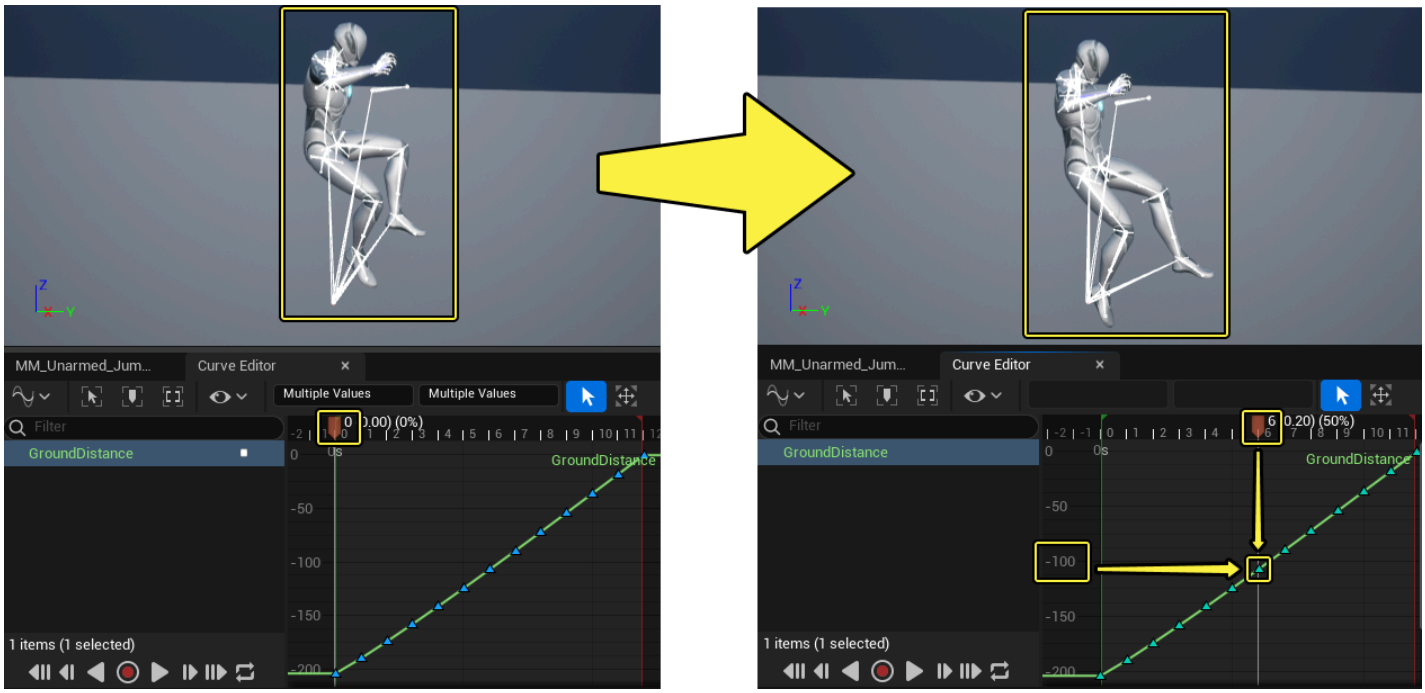
Double-click the curve to open it in the Curve Editor. In the Curve Editor, you have access to detailed information about the curve and editing tools to tweak or adjust the curve.

The example below shows a Ground Distance curve graphing the animation sequence's poses on the x-axis, and the units of vertical displacement on the y-axis, where y = 0 is the

level's ground. In the example curve, triangles representing keyframes are consistently displaced along the curve trajectory (the green line). However, a pose can be selected at any point on a trajectory, not only at the keyframe intersections. Using Distance Matching, the Animation Blueprint searches the Animation Curve to find the indexed value, denoted by the distance variable, and outputs the corresponding Animation Pose.



Below is a manual demonstration of this process. Given a distance value of -100 units from the ground, a pose is located around the 6th keyframe on the curve and selected for playback.



A Distance Matching Node will use an Animation Distance Curve to select an animation pose to adjust the animation sequence playback. With Distance Matching, animation playback

speed can be driven to match character movement speed, which reduces the need to fine tune animations and allows for the character's speed to be altered without disrupting animation playback.

Below is a practical implementation of Distance Matching in action, within the Lyra sample project.

Distance Matching Workflow

This workflow example demonstrates a basic implementation of Distance Matching. Using the **Sequence Evaluator** node to evaluate a landing animation, the Distance Matching logic selects an animation pose based on the remaining distance to the ground from the character's location using an **Animation Distance Curve**.

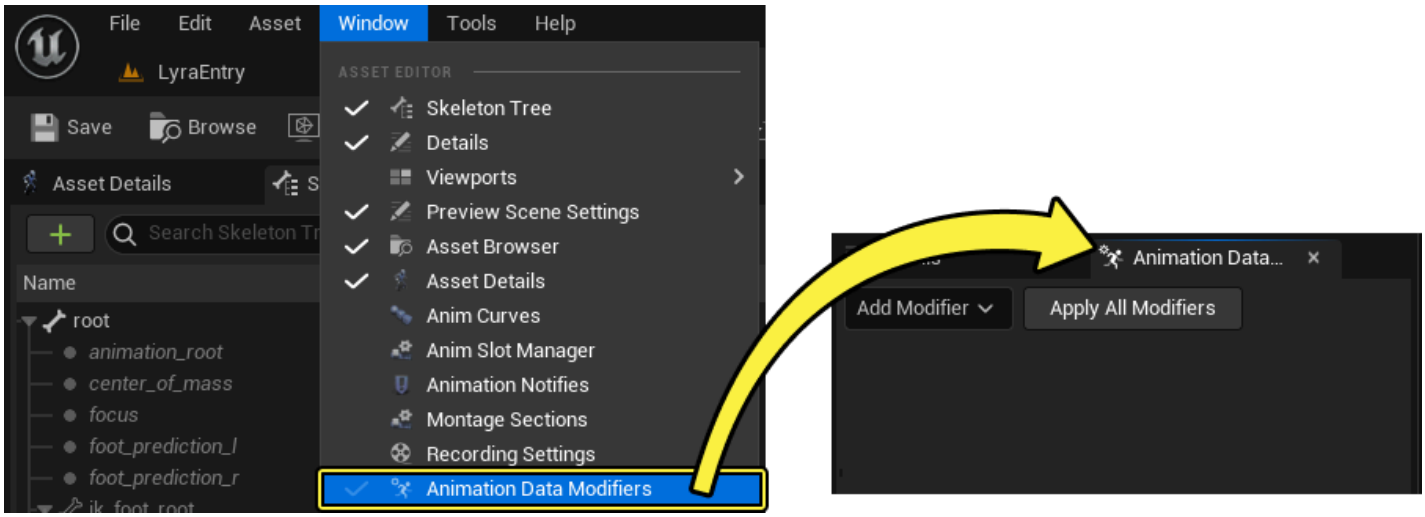
Curve Generation

Distance Matching requires an Animation Curve graphing animation poses with the distance to or from a target. This curve can be referenced to adjust the animation playback pose according to a dynamic distance variable.

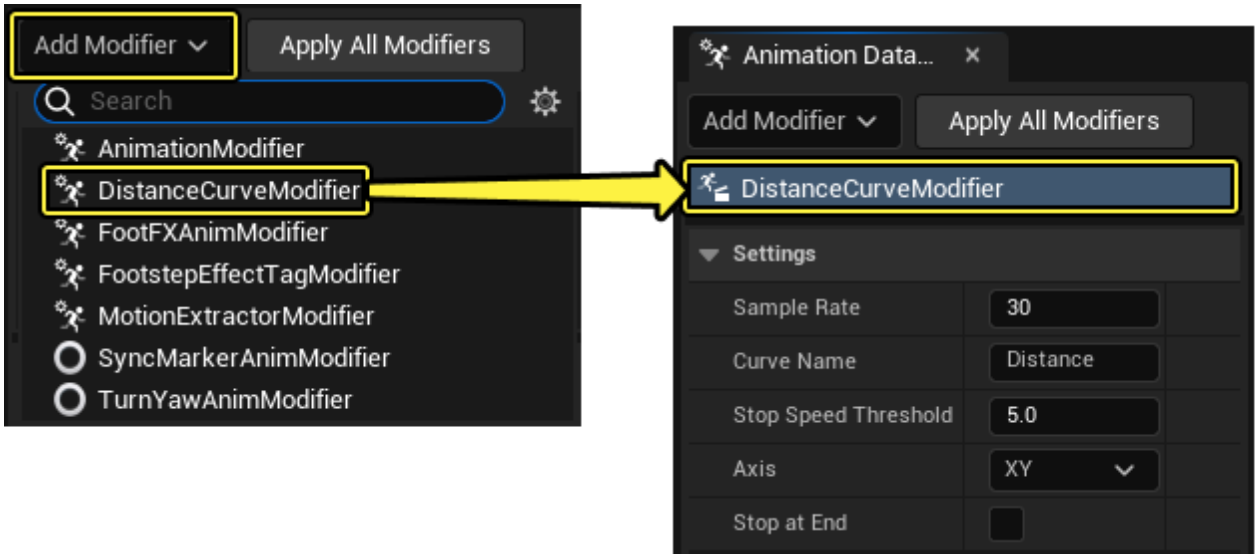
For example, instead of the 9th keyframe transitioning to the 10th keyframe based on a unit-of-time threshold, the transition occurs based on a unit-of-distance threshold.

This curve can be generated form an Animation Sequence with Root Motion enabled.

To generate a curve, open an Animation Sequence in the Animation Sequence Editor. Navigate to **Menu Bar > Windows,** and in the **Asset Editor** section, select **Animation Data Modifiers** to open the window.



Select **Add Modifier** and choose the **Distance Curve Modifier** option from the drop down menu.



In the **Distance Curve Modifier Settings**, set the parameters for generating the curve.

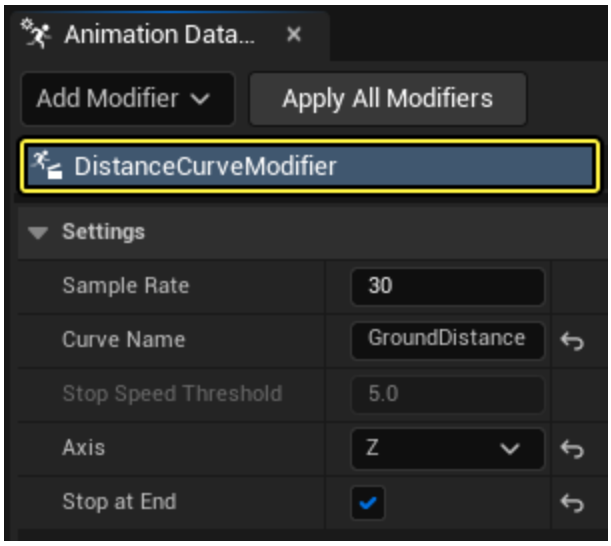
Below is a list of the **Distance Curve Modifier Properties** and their functions.

Property	Description
Sample Rate	Determines how many curve samples to generate for the sequence. Higher values increase the samples and the distance curve precision. A value of 30 covers most use cases.
Curve Name	Defines the name of the generated curve.
Stop Speed Threshold	Defines the threshold when the character has stopped moving or reached the animation end point. A value of 0 is the default, but a character doesn't always reach 0 before the animation is finished. This property can be used to tweak the end threshold for smoother transitions. For example, in a pivot animation, the character changes direction but may never fully stop. Tweak this threshold to specify how slow the character must be to be "stopped".
Axis	Determines what axis / axes of motion to consider. "Z" is useful for land animations. "XY" is useful for locomotion transitions (stop, start, pivot).
Stop at End	If enabled, the curve only declares the character stopped on the last frame of the Animation Sequence. Enable this property if the character is or can be stopped before the last frame of Animation Sequence.

For the demonstration, change the **Axis property** to the **Z** axis, since the animation is Distance Matched according to a change in height.

Enable the **Stop at End** setting.

Name your curve using the **Curve Name** property, for later reference.



Leave other settings as the default option.

Click **Apply All Modifiers** at the top of the window to generate the new curve.

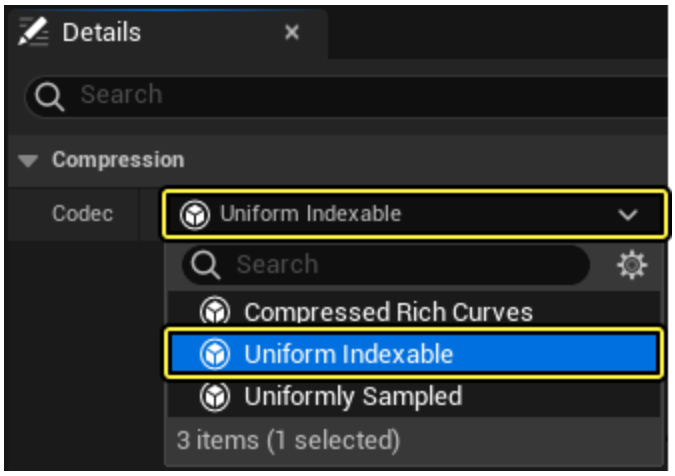
Now that you have the Animation Distance Curve, you can move back and forth along the curve using the playhead in the Curve Editor window, to view the Animation Pose output at different distance values from your target. For this demonstration, the character extends their legs to meet the approaching ground.

Compression

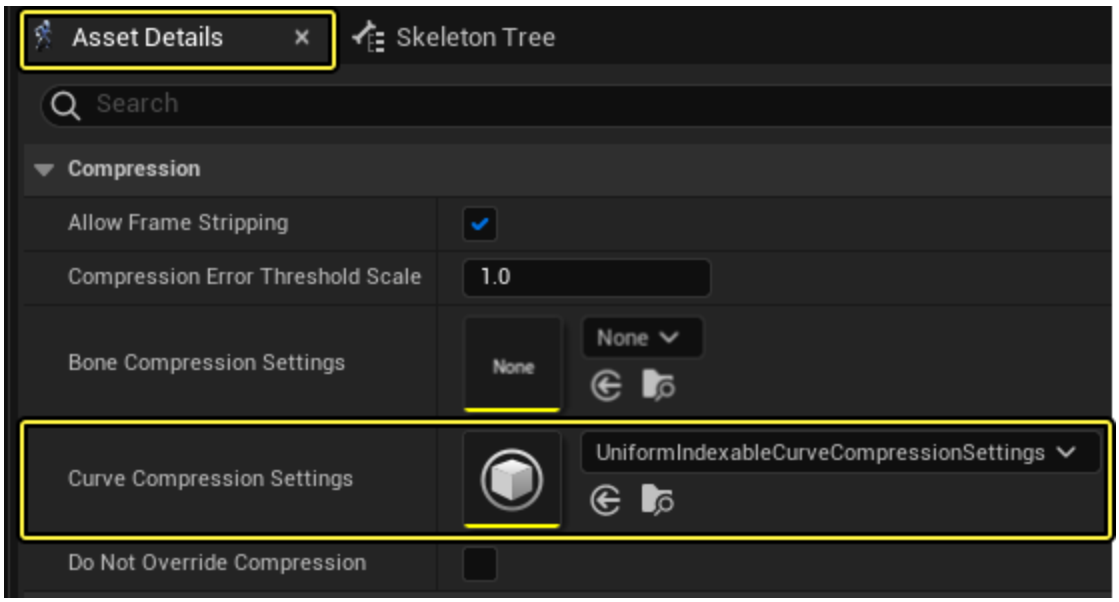
Distance Matching an animation sequence requires a **Uniform Indexable Animation Compression Setting** asset in place of default compression methods, to read the distance curve at runtime.

To apply this compression, create a new **Animation Compression Settings Asset**. Right-click in the **Content Browser** and navigate to the **Animation** dropdown menu under **Create Advanced Asset**, then select **Curve Compression Settings**.

Double-click the new **Compression Asset** to open the **Details** panel. Under **Compression**, select **Uniform Indexable** for the **Codec** option.

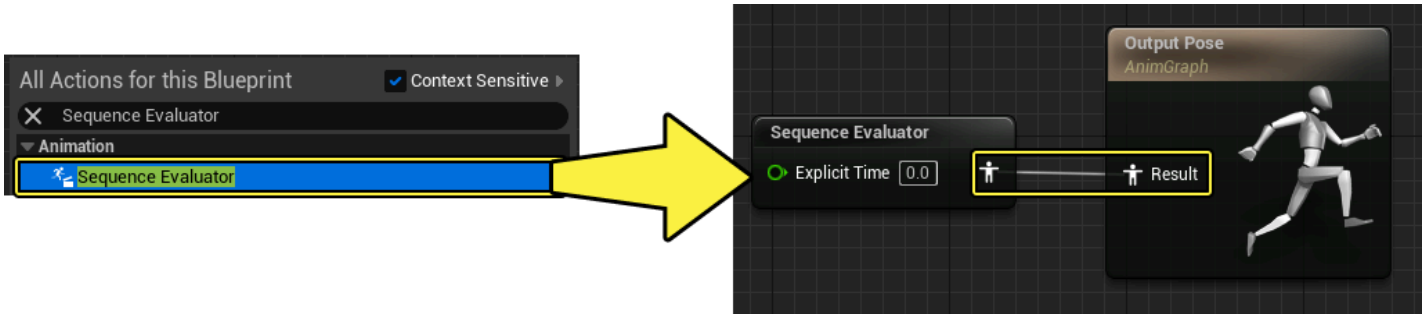


Within the Animation Sequence you are Distance Matching, assign the new **Curve Compression Settings** asset in the **Details** panel under **Compression**.



Animation Blueprint

In the character's Animation Blueprint **AnimGraph**, create a **Sequence Evaluator** node and connect it to the **Output Pose** node.

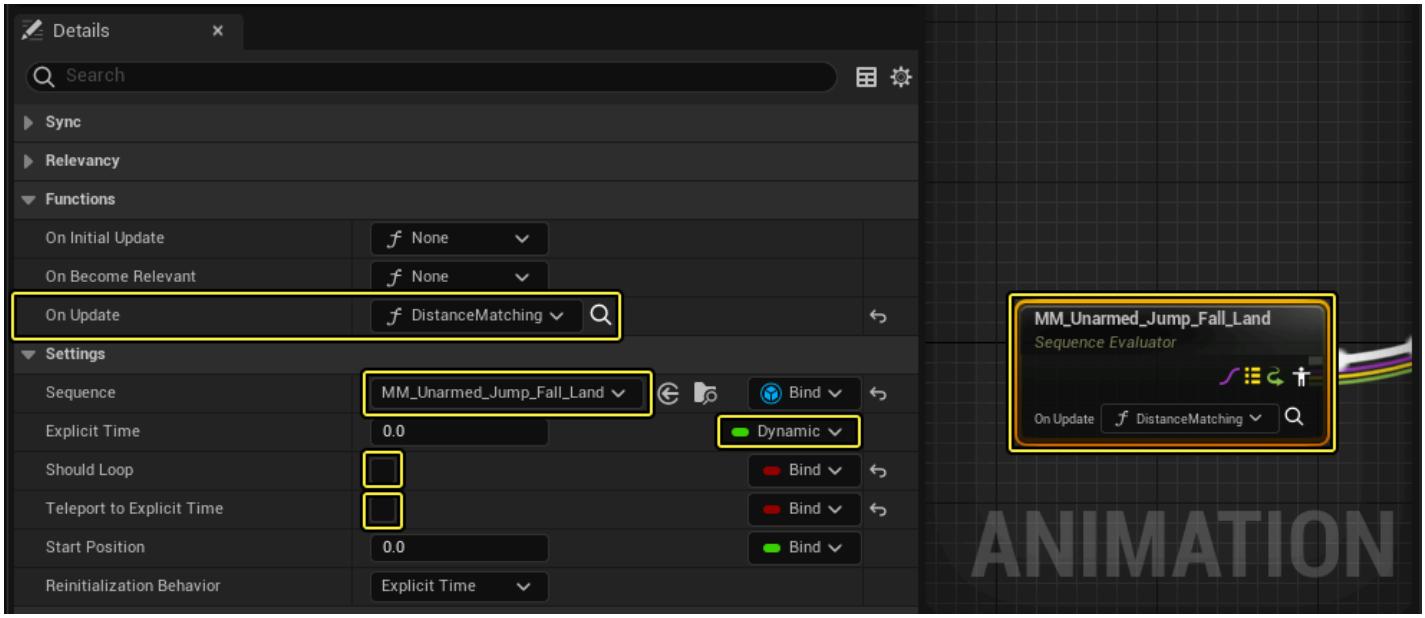


Open the Sequence Evaluator node's **Details** panel to define the Animation for Distance Matching.

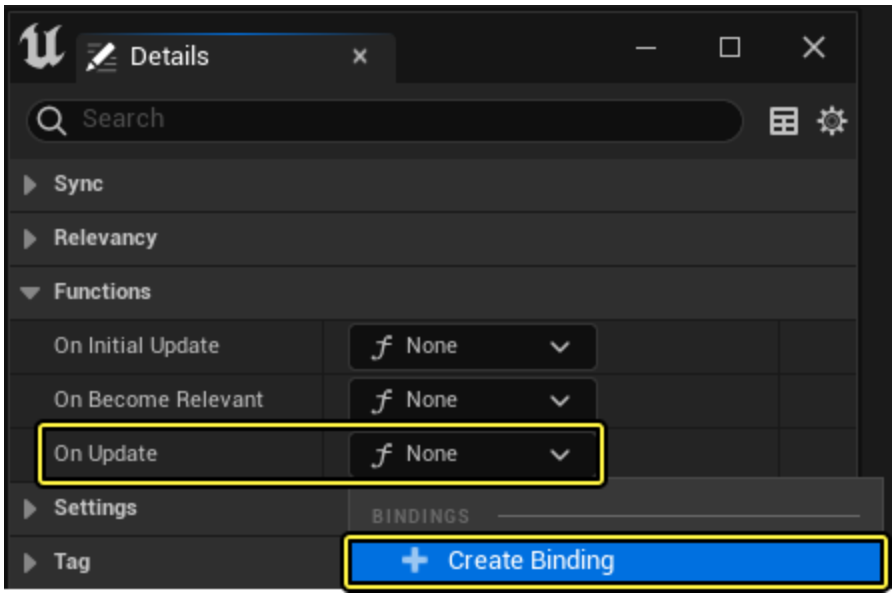
Next, in the **Explicit Time** property, open the pin's **drop down menu**, and toggle the **Dynamic Value** option in the **Pin** category. Toggling the **Dynamic Value** option allows you to set the value using the Anim Node Function. It is safe to hide this pin in the Blueprint.

Disable **Should Loop** and **Teleport to Explicit Time**.

Set the **Start Position** to 0 and **Reinitialization Behavior** to rely on **Explicit Time**.



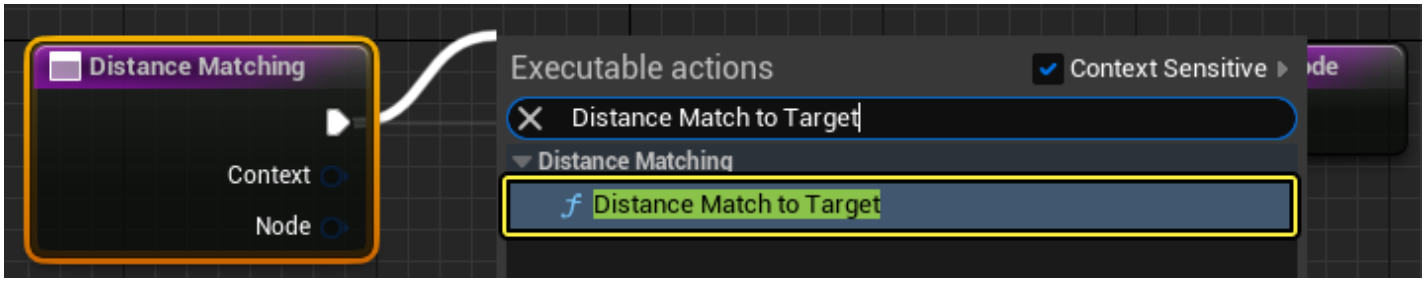
After you define the properties of the Sequence Evaluator node, create an Animation Node Function on the Sequence Evaluator node to drive the logic. To create this internal function, navigate to the **Functions** section in the Sequence Evaluator node's **Details** panel. In the **On Update** property, open the **dropdown menu** and select **+Create Binding**.



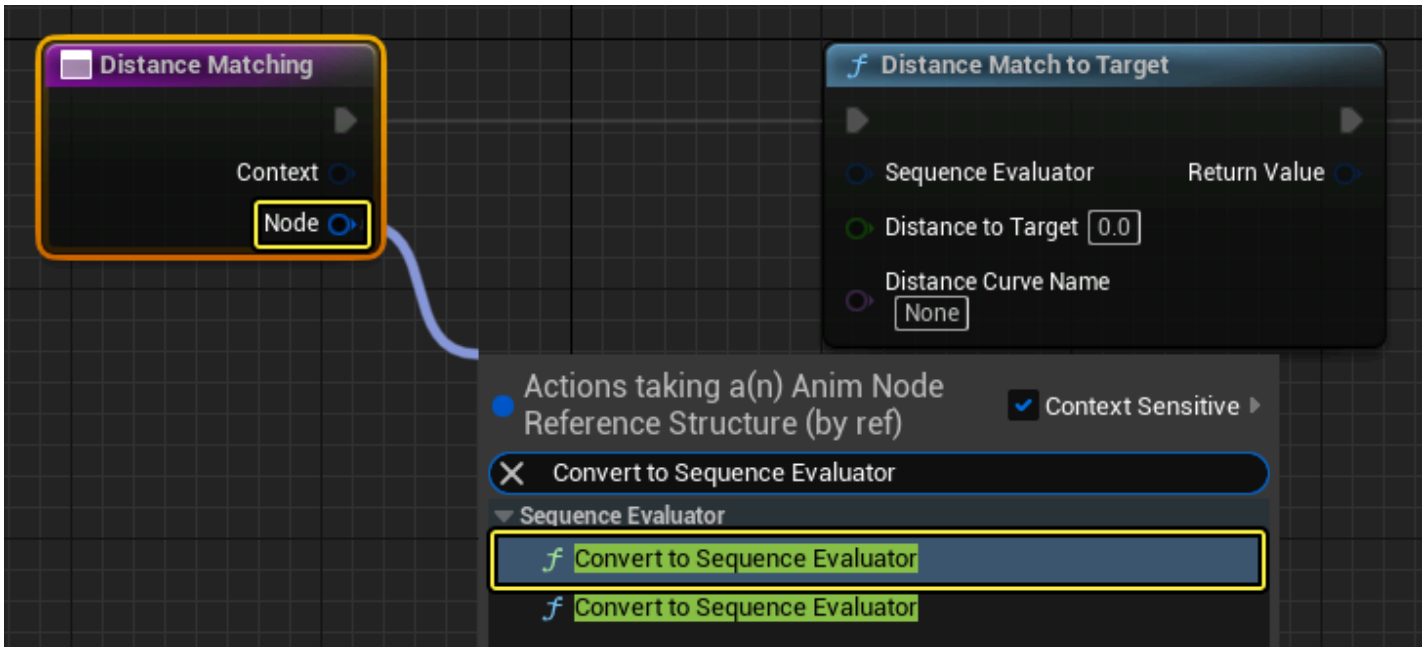
You can rename the function in the **Details** panel. In this example, the function is named "Distance Matching".

The function opens in a new tab in the **Blueprint Editor Viewport** window.

Between the **In** and **Out** nodes of the function, create a **Distance Match to Target** node.



Next, drag off of the **Distance Match to Target** node's **Node output pin** and create a **Convert to Sequence Evaluator** float function (green) node.



Connect the **Convert** to **Sequence Evaluator** node's **Sequence Evaluator output pin** to the **Distance Match To Target**'s **Sequence Evaluator input pin.**

Call an instance of the **Float Variable** that calculates the distance from the target. In this example, the variable measures the distance from the character to the ground, and is called **Distance to Ground**.

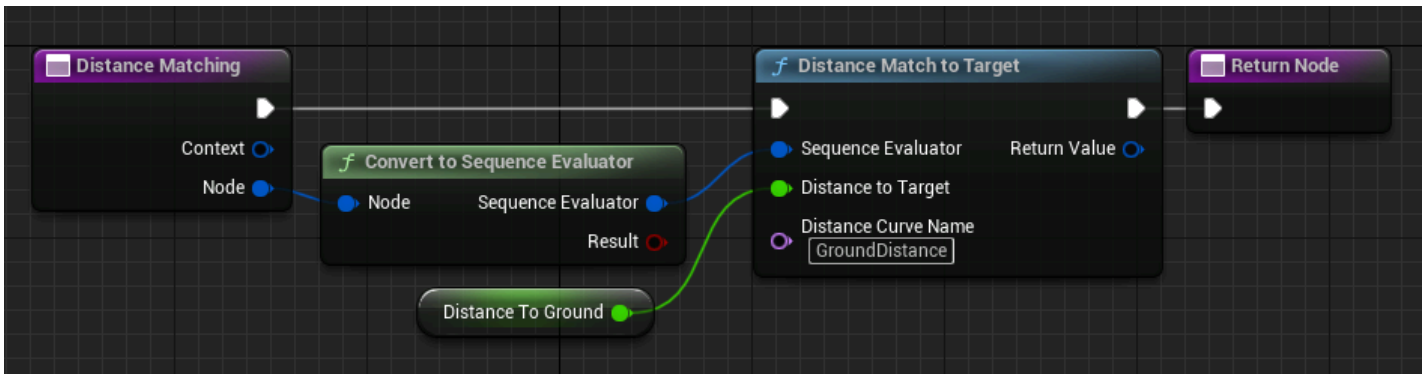
On the **Distance Match to Target** node, define your **Distance Curve Name**.

Save and **Compile** the Blueprint and you can now see the results of Distance Matching in the **Animation Preview Viewport**.

Results

While running the animation, in the **Scene Setup** section of the **Viewport Settings**, you can adjust the **Floor Height Offset** property to show Distance Matching in action. With Distance Matching enabled and implemented the Sequence Evaluator node selects the animation poses from the curve as you alter the distance between the character and the ground. This creates a more natural landing animation, where the character dynamically reacts to the approaching ground. With distance matching, the animation sequences can also adjust to changes in the character's falling speed, without the need to reanimate the sequence to match an adjusted fall rate.

Below is the full Anim Node function Blueprint used in the example workflow to implement Distance Matching within the Sequence Evaluator node.

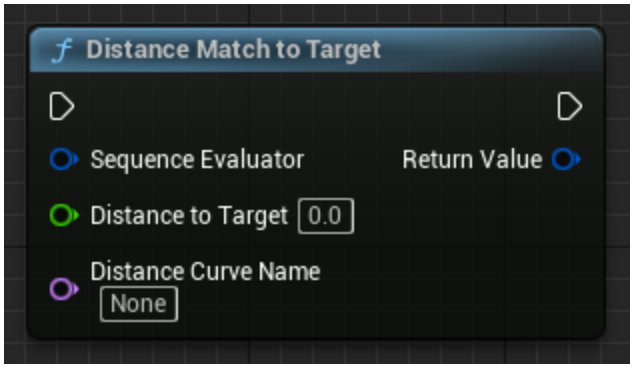


Distance Matching Node Reference

Below is a list of the Distance Matching nodes that operate in Anim Node Functions.

Distance Match to Target

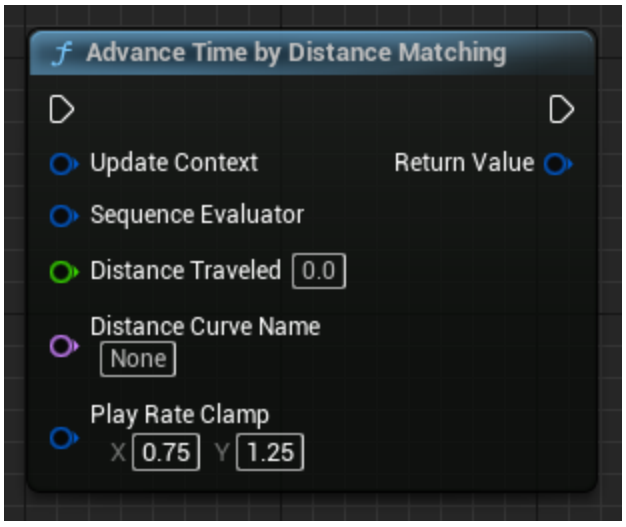
The **Distance Match To Target** node uses an Animation Curve and a distance variable to select animation poses, to scale the animation playback speed to match character speed.



Under the Animation Blueprint section of the workflow demonstration, there is an implementation of the Distance Match To Target node in an Anim Node Function.

Advance Time By Distance Matching

The **Advance Time By Distance Matching** node advances a connected **Sequence Evaluator** node forward by the distance the character travels each frame.



This node requires an Animation Distance Curve graphing the distance traveled by the Root Bone to select animation poses from, and a variable measuring the distance traveled.

Set Playrate to Match Speed

The **Set Playrate to Match Speed** node sets the play rate of an **Animation Sequence Player** node so that the speed of the animation matches the character's movement speed, assuming the character has consistent movement speed.