

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Animating Characters and Objects

/ Skeletal Mesh Animation System

/ Animation Assets and Features

/ Motion Matching

Motion Matching

Use Motion Matching to create responsive animation systems, that select animation poses from a database, to match a dynamicsystem query at runtime.

Motion Matching in Unreal Engine is a query-based animation pose selection system. Contained within the **Pose Search** plugin, you can use the Motion Matching [Animation Blueprint](#) node as a dynamic alternative to [State Machines](#), or [Blendspaces](#). Unlike traditional animation systems, Motion Matching can make informed animation pose selections from a set of animation data, to create a responsive and reactive animation system, without the need to set up transition or blending logic between [Animation Sequences](#).



The Motion Matching node queries channels from the character via the schema (such as a Bone's position and velocity), and makes informed animation pose selection, from a set of animation data stored in a **Database** asset, to match the query. Motion Matching character animation results are more reactive and responsive to gameplay systems than traditional State Machines or blend logic.

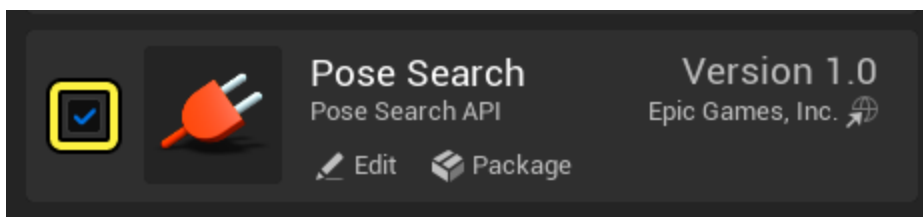
With Motion Matching, you can increase your character's animation fidelity, by just adding more animation data to the Database asset. With more animation data, the Motion Matching node has access to more data to select from, without the need to alter or implement additional transition logic.

Unreal Engine's Motion Matching system also contains a suite of settings and debugging tools that you can use to tune your animation system to match the goals of your project. With these tools, you can observe selection choices, edit the query process, and set selection criteria weights.

This document provides an overview of Motion Matching in Unreal Engine as well as an example workflow of how to set up a character's locomotion animation system using Motion Matching.

Prerequisites

- Enable the **Pose Search** [Plugin](#). Navigate in the **Menu Bar** to **Edit > Plugins** and locate the in the **Animation** section, or use the **Search Bar**. Enable the Plugin and restart the editor.



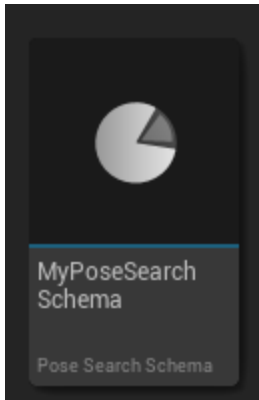
- Your project contains a Character, with a set of locomotion (walking, running and turning) animations. If you do not have these, you can download the Game Animation Sample Project, from the Unreal Engine Marketplace. For more information about this sample project see the [Game Animation Sample Project](#) documentation.

Motion Matching Setup

Here you can follow an example workflow about how to set up a simple locomotion based Motion Matching system in your project. For a more detailed and advanced setup, see the [Game Animation Sample project](#) documentation.

Create a Pose Search Schema Asset

A Pose Search Schema asset stores your motion matching configuration and query settings. This asset is used to link your animation databases and query system to the Motion Matching node, and defines the data that will be used to query in order to make animation selections.



To create a Pose Search Schema asset, **right-click** in the **Content Browser** to **(+) Add** and select **Animation > Motion Matching > Pose Search Schema**.

Q Start typing to search

GET CONTENT

- Import to /Game...
- Add Feature or Content Pack...
- Add Quixel Content

FOLDER

- New Folder

CREATE BASIC ASSET

- Blueprint Class
- Level
- Material
- Niagara System

CREATE ADVANCED ASSET

Animation

Artificial Intelligence

Audio

Blueprint

Cinematics

Editor Utilities

Foliage

FX

Gameplay

Input

Interchange

Material

Media

Miscellaneous

Paper2D

Physics

Texture

Tool Presets

User Interface

World

Q Start typing to search

- Aim Offset
- Animation Blueprint
- Animation Composite
- Animation Layer Interface
- Animation Montage
- Blend Space
- Contextual Anim Scene
- Mirror Data Table
- Pose Asset
- Variable Frame Stripping Settings

Advanced

Control Rig

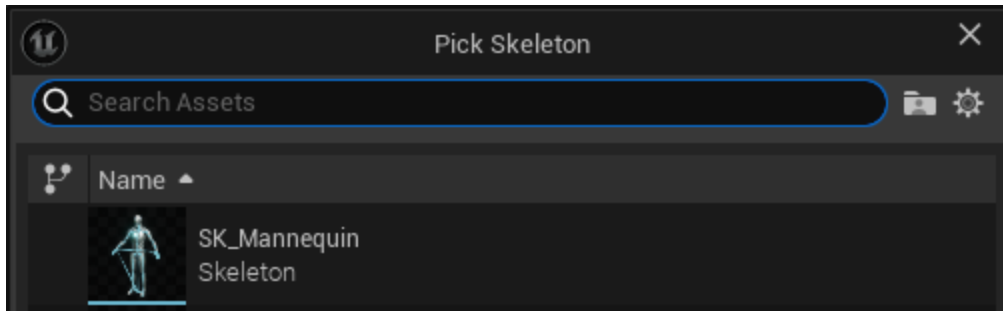
Legacy

Motion Matching

Retargeting

- Pose Search Database
- Pose Search Normalization Set
- Pose Search Schema

After creating a new Pose Search Schema asset, you will be prompted to select your character's skeleton that you are building the Motion Matching system for.



New Pose Search Schema assets come preconfigured for a locomotion setup, with two channels, the [Trajectory Channel](#) and the [Pose Search Channel](#). With the Trajectory channel you can set points on the Motion Trajectory Component to sample in order to make pose selections. The default settings will work for this example workflow, however you can add additional samples using (+) **Add** in the **Samples** property, to provide more usable data for the Motion Matching system to make selections.

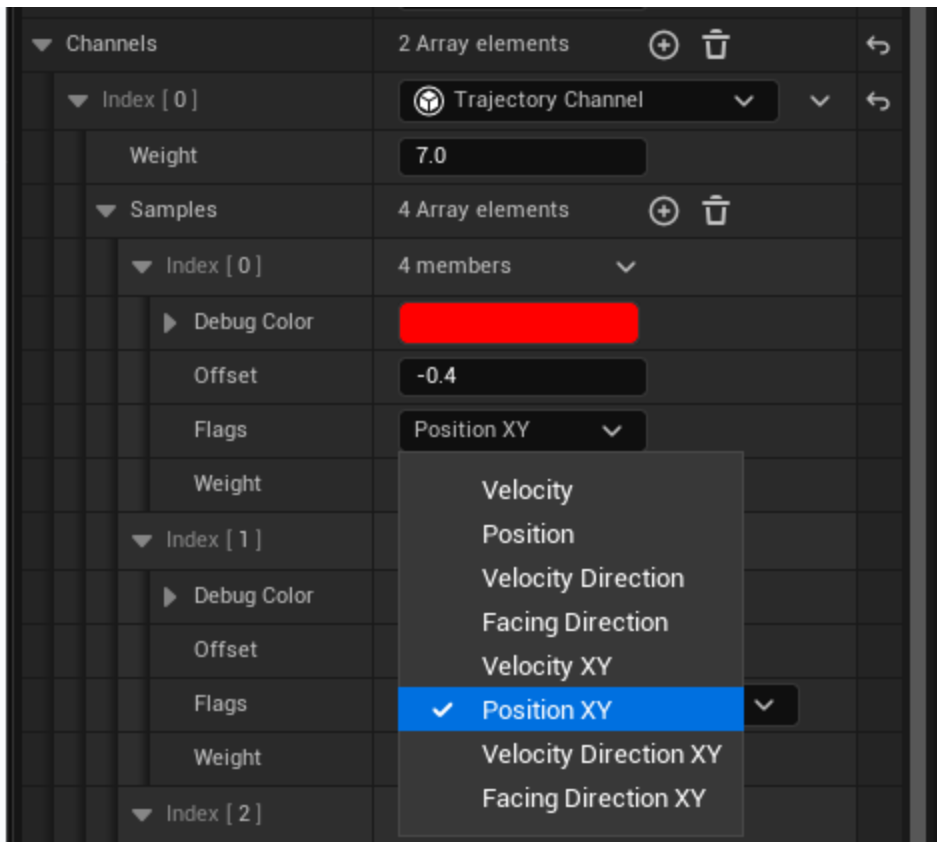


It is recommended that you use as few samples as possible. While more samples can provide the Motion Matching node with more accurate data, it can distort the system's results.

▼ Channels	2 Array elements	⊕	🗑️	↶
▼ Index [0]	Trajectory Channel ▼ ↶			
Weight	7.0			
▼ Samples	4 Array elements	⊕	🗑️	
▼ Index [0]	4 members	▼		
▶ Debug Color	<div></div>			
Offset	-0.4			
Flags	Position XY ▼			
Weight	0.4			
▼ Index [1]	4 members	▼		
▶ Debug Color	<div></div>			
Offset	0.0			
Flags	Velocity XY Facing Direction XY ▼			
Weight	2.0			
▼ Index [2]	4 members	▼		
▶ Debug Color	<div></div>			
Offset	0.35			
Flags	Position XY Facing Direction XY ▼			
Weight	0.7			
▼ Index [3]	4 members	▼		
▶ Debug Color	<div></div>			
Offset	0.7			
Flags	Velocity XY Position XY Facing Dire- ▼			
Weight	0.5			

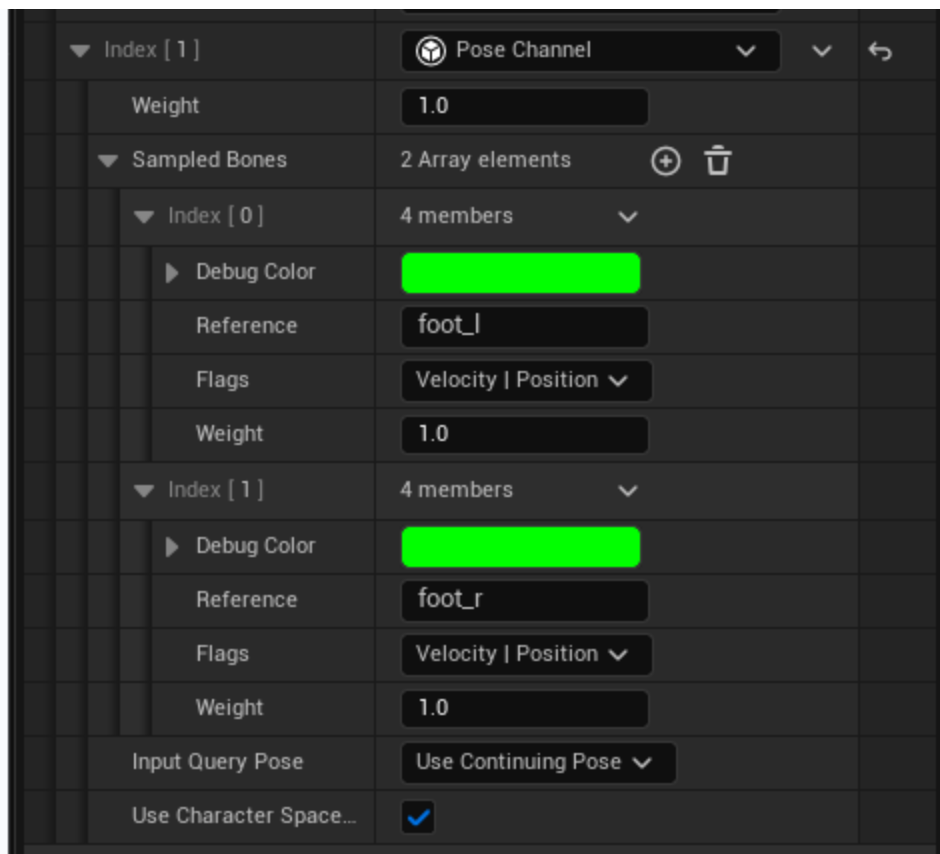
Trajectory Channel

Each Index within the Trajectory Channel, you can assign Flags, to determine what type of data the sample point will look for when querying the Motion Trajectory Channel. The default values will work for this workflow example. For more information about the Trajectory Channel's settings and Flags, see the [Trajectory Channel Reference](#) section.



Pose Channel Channel

With the **Pose Channel**, you can set bone definitions for the Motion Matching system to match poses with the query system. The default asset comes pre-configured with bone definitions for the Mannequin Skeleton's foot bones `foot_l` and `foot_r`. If you are using a different skeleton replace these bone definitions with your skeleton asset's left and right foot bones.



You can also set Flags for each sampled bone to determine what type of data from the sampled bone the motion matching system will look at in order to make selections. For more information about the Pose Channel's settings and the difference Flags, see the [Pose Channel](#) section.



The Pose Channel can use any bone as a reference point. For a locomotion motion matching animation system, you will want to select a pair of bones that will be useful to track motion data related to motion, such as the characters feet. In an alternative system, such as a climbing system, you will want to select bones that make more sense in containing valuable data, such as the characters' hand bones.




More channels can be added to the Motion Database Config asset, but the more channels and samples within the channels you have, the more performance it will take to run.

For this workflow example, the default values will provide good locomotion results.

Pose Search Schema Asset Reference

Here you can reference a list of the Pose Search asset's properties and settings.

Property	Description
Skeleton	Here you can define the Skeleton asset for your character that will use the Motion Matching node.
Mirror Data Table	Here you can set a Mirror Data Table asset to provide mirrored animation sequences for the Motion Matching node to use when making animation selections.
Data Preprocessor	<p>Set the type of operation performed to the full pose features dataset. You can select from the following options:</p> <ul style="list-style-type: none">• None: This option will not apply any data preprocessing.• Normalize: This option will normalize all data against its deviation. Additionally, all weights will be normalized to be a unitary vector value.• Normalize Only by Deviation: This option will only normalize all data against its deviation, without normalizing weight values.
Sample Rate	Set the update rate at which the animation data is sampled in the database. The higher the Sample Rate the more refined animation pose searches will be, but will increase the memory cost of the system.
Channels	<p>Here you can add and set Channels for the Motion Matching system to use when Querying, Searching, and Matching animation poses at runtime. Channels itemize the cost breakdown of the Pose Search Schema asset into simpler parts such as the position or velocity of samples bones, or phases of limbs. The total cost of a query against an indexed database pose will be the sum of the combined channel costs. You can add a channel using (+) Add and selecting a channel from the drop down menu. Adding additional channels will detail your search of the motion matching system but will require more memory and performance cost.</p> <div></div>

Property	Description
	<p>It is recommended to use as few channels as necessary to achieve your animations system's goals.</p> <p>For more information about the individual channels and their specific properties, see the Channels section.</p>
Number of Permutations	Set the number of times the animation assets from the associated database will be indexed while making selections.
Permutations Time Offset	<p>Set a value that will represent the starting offset of the PermutationTime from the Sampling Time of the first permutation. Subsequent permutations will use the flowing formula to determine the Permutation Time Offset:</p> $\text{PermutationTime} = \text{SamplingTime} + \frac{\text{PermutationTimeOffset} + \text{PermutationIndex}}{\text{PermutationSampleRate}}$
Add Data Padding	When enabled, a padding channel will be added to the Motion Database Config asset to make sure the data is aligned 16 bytes and padded to facilitate performance improvements at cost of eventual additional memory.
Inject Additional Debug Channels	When enabled, channels will be asked to inject additional channels into this schema. The original intent is to add UPoseSearchFeatureChannel_postions(s) to help with the complexity of the debug drawing (the database will have all the necessary positions to draw lines at the right location and time).

Channels

Here you can reference a list of the available channels for a Motion Database Schema asset.

Group

You can use the **Group Channel** to group other Pose Search Schema asset channels into groups in order to run operations together. After creating the Group Channel you can create additional channels within its Sub-Channel property using **(+) Add**.

Heading

You can use the Heading Channel to query your systems heading direction. This can be useful when you want to match a specific bone's orientation components, expressed as a heading axis, such as only the bone's Horizontal or Vertical movement.

Here you can reference a list of the Heading Channel's properties and a description of their function:

Property	Description
Debug Color	Here you can set a debug drawing color that will render in the viewport to assist with debugging your motion matching system.
Bone	Here you can assign a bone from your character's skeleton that will be used in the query to make pose selections. You can select a bone from the skeleton's hierarchy using the drop down menu.
Origin Bone	Set a Bone from the Skeleton that will serve as the Origin Bone to determine the heading direction from. You can select a bone from the character's skeleton using the drop down hierarchy menu.
Weight	Set a weight value that will be used to weigh this channel's influence on the output pose against the other channels in your Motion Database Config asset.
Sample Attribute ID	Set a value that will be used as the Sample Attribute ID. If this value is greater than <input type="text" value="0.0"/> , all the animations contained in the Pose Search Database associated with this Pose Search Schema asset, contained in this channel, are expected to have the Pose Search: Sampling Attribute ID Notify State, and the Sampling Attribute property defined. The value set in this property will be used as the data source instead of the Bone channel. The Sampling Attribute property's values will be converted into Origin Bone space.

Property	Description
Sample Time Offset	Here you can set a value that will be used as an offset for the sample time.
Origin Time Offset	Here you can set a value that will be used to offset the Origin Time. The data relative to the Sampling Time associated with this channel's Origin Bone (Root or Trajectory Bone) will be offset by the value of this property in seconds. For Example, if the Origin Bone is set to the character's head bone, the Sample Time Offset is set to <input type="text" value="0.5"/> , and the Origin Time Offset is set to <input type="text" value="0.5"/> , this channel will try to match the future heading of the character's head bone <input type="text" value="0.5"/> seconds into the future, relative to the future root bone at the same time.
Heading Axis	Set the axis (X , Y , or Z) that will be used to query the sampled Bone property's motion.
Input Query Pose	Here you can set the method used to input the queried pose. You can select from the following methods: <ul style="list-style-type: none"> • Use Character Pose: Uses the character's current pose to input the queried pose. • Use Continuing Pose: Uses the character's continuing pose to input the queried pose. • Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.
Component Stripping	Here you can select the specific axis of motion to strip from the queried data. Selecting None performs no component stripping, selecting Strip XY will only observe the X and Y axis of motion, matching along the horizontal plane, and selecting Strip Z will only observe motion on the Z axis, only matching along the height motion of the query.
Permutation Time Type	Here you can select the type of permutation time to use when sampling animations. You can select from the following options: <ul style="list-style-type: none"> • Use Sample Time: This type samples the Bone and Origin Bone at the same sample time, in addition to the eventual

Property	Description
	<p>Sample Time Offset of the Bone. It is defined as the current animation evaluation time.</p> <ul style="list-style-type: none"> • Use Permutation Time: This type samples the Bone and Origin Bone at the same permutation time, in addition to the eventual Sample Time Offset of the Bone. It is defined as SampleTime (as UseSampleTime) $\left(+ \text{Schema} \rightarrow \text{PermutationTimeOffset} + \frac{\text{PermutationIndex}}{\text{Schema} \rightarrow \text{PermutationSampleRate}} \right)$ where PermutationIndex is in range $[0, \text{Schema} \rightarrow \text{Number of Permutations}]$. • Use Sample to Permutation Time: This type evaluates the Bone at the sample time, in addition to the eventual Sample Time Offset for the Bone, and the Origin Bone is evaluated at Permutation Time.
Normalization Group	<p>Here you can set a normalization group to normalize the channel's query against other Heading Channels. If set, all the channels of the same class with the same cardinality, and the same Normalization Group, will be normalized together. For example, in a locomotion database of a character holding a weapon. Containing non-mirrabable animations, you may still want to normalize left foot and right foot positions and velocity.</p>

Padding

The Padding channel is used to add additional padding data into your Motion Matching database to facilitate performance improvements at cost of eventual additional memory. The channel is automatically injected into the Pose Search Schema if the Add Data Padding property is enabled and set to 16 bytes. You can set the number of permutations the channel pads the data using the **Padding Size** property.

Phase

Here you can reference the Phase channel's properties and a description of their functionality:

Property	Description
Debug Color	Here you can set a debug drawing color that will render in the viewport to assist with debugging your motion matching system.
Bone	Here you can assign a bone from your character's skeleton that will be used in the query to make pose selections. You can select a bone from the skeleton's hierarchy using the drop down menu.
Weight	Set a weight value that will be used to weigh this channel's influence on the output pose against the other channels in your Motion Database Config asset.
Input Query Pose	<p>Here you can set the method used to input the queried pose. You can select from the following methods:</p> <ul style="list-style-type: none"> • Use Character Pose: Uses the character's current pose to input the queried pose. • Use Continuing Pose: Uses the character's continuing pose to input the queried pose. • Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.

Pose

You can use the Pose channel to calculate the character's position in character space, rather than in global space. This channel is useful when the movement you want to query is local to that character's position, rather than global movement.

Here you can reference the Pose Channel's properties and a description of their functionality:

Property	Description
Weight	Set a weight value that will be used to weigh this channel's influence on the output pose against the other channels in your Motion Database Config asset.

Property	Description
Sampled Bones	<p>Here you can define a list of bones the Pose Channel will sample based on the associated Flags, to match poses against the queries system. You can add sampled bone definitions using (+) Add. After adding an index, you can set the following properties:</p> <ul style="list-style-type: none"> • Debug Color: Set the color that will be used to render a debug drawing of the Reference sampled bone at runtime in the viewport. • Reference: Set a bone from the character's skeleton hierarchy using the drop down menu. • Flags: Here you can toggle Flags that will determine what type of data will be referenced from the sampled bones. You can toggle as many flags as needed to meet your project's goals. You can toggle from the following list: <ul style="list-style-type: none"> ◦ Velocity ◦ Position ◦ Rotation ◦ Phase • Weight: Set a weight value that will be used to weigh this Sampled Bones index's influence on the Channel.
Input Query Pose	<p>Here you can set the method used to input the queried pose. You can select from the following methods:</p> <ul style="list-style-type: none"> • Use Character Pose: Uses the character's current pose to input the queried pose. • Use Continuing Pose: Uses the character's continuing pose to input the queried pose. • Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.
Use Character Space Velocities	<p>When enabled, velocities will be calculated from the positions in character space, otherwise they will be calculated using global space positions.</p>

Position

You can use the Position Channel to query the position of a bone relative to another in your character's skeleton. This can be useful when trying to make pose selections based on a bone's displacement, or position relative to another.

Here you can reference a list of the Position Channel's properties and a description of their functionality:

Property	Description
Debug Color	Here you can set a debug drawing color that will render in the viewport to assist with debugging your motion matching system.
Bone	Here you can assign a bone from your character's skeleton that will be used in the query to make pose selections. You can select a bone from the skeleton's hierarchy using the drop down menu.
Origin Bone	Here you can assign a bone from the character's skeleton that will be used as a reference point, to calculate the Bone's positional change. You can select a bone from the skeleton's hierarchy using the drop down menu.
Weight	Set a weight value that will be used to weigh this channel's influence on the output pose against the other channels in your Motion Database Config asset.
Sample Time Offset	Here you can set a value that will be used as an offset for the sample time.
Input Query Pose	<p>Here you can set the method used to input the queried pose. You can select from the following methods:</p> <ul style="list-style-type: none">• Use Character Pose: Uses the character's current pose to input the queried pose.• Use Continuing Pose: Uses the character's continuing pose to input the queried pose.• Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.

Property	Description
Component Stripping	Here you can select the specific axis of motion to strip from the queried data. Selecting None performs no component stripping, selecting Strip XY will only observe the X and Y axis of motion, matching along the horizontal plane, and selecting Strip Z will only observe motion on the Z axis, only matching along the height motion of the query.
Permutation Time Offset	Set a value that will represent the starting offset of the PermutationTime from the Sampling Time of the first permutation. Subsequent permutations will use the flowing formula to determine the Permutation Time Offset: $\frac{\text{PermutationTime} = \text{SamplingTime} + \text{PermutationTimeOffset} + \text{PermutationIndex}}{\text{PermutationSampleRate}}$

Sampling Time

The Sampling Time Channel is used for debugging purposes, to augment the features data with the sampling time. By default this Channel's **Weight** is set to , so it will not be considered during searches.

Trajectory

The Trajectory Channel is used for sampling a character's movement trajectory and is designed to be used in conjunction with the **Character Trajectory** blueprint component. Using the **Weight** property you can set the amount of influence the Trajectory Channel has over the final output pose. With the Samples property you can add samples from points in time on the Character Trajectory component in which to sample the characters movement model.

Here you can reference a list of the Samples' properties and a description of their functionality:

Property	Description
Debug Color	Here you can set a debug drawing color that will render in the viewport to assist with debugging your motion matching system.
Offset	Here you can set when the character's movement sample should be taken from the Character Trajectory component. Positive values look forward in time, while negative values look backwards in time.
Flags	Here you can toggle which values from the character's movement should be taken at the location along the Character Trajectory component, specified by the Offset property. You can toggle any number of values from the drop down list.
Weight	Set a value that will be used to weight the Samples' influence over the final output pose selection. This value is only used to weigh against other Samples in the index, and not against other channels.

Velocity

You can use the Velocity Channel to query poses based on the character's velocity either in Character Space or in Global Space.


Here you can reference a list of the Velocity Channel's properties and a description of their functionality:

Property	Description
Debug Color	Here you can set a debug drawing color that will render in the viewport to assist with debugging your motion matching system.
Bone	Here you can assign a bone from your character's skeleton that will be used in the query to make pose selections. You can select a bone from the skeleton's hierarchy using the drop down menu.

Property	Description
Weight	Set a weight value that will be used to weigh this channel's influence on the output pose against the other channels in your Motion Database Config asset.
Sample Time Offset	Here you can set a value that will be used as an offset for the sample time.
Input Query Pose	<p>Here you can set the method used to input the queried pose. You can select from the following methods:</p> <ul style="list-style-type: none"> • Use Character Pose: Uses the character's current pose to input the queried pose. • Use Continuing Pose: Uses the character's continuing pose to input the queried pose. • Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.
Use Character Space Velocities	When enabled the velocity of the specified Bone will be calculated in character space rather than globally. This is helpful when you are querying a localized movement, rather than the movement of the character in world space.
Normalize	When enabled the velocity values will be normalized to a maximum value of <input type="text" value="1"/> . This property can be helpful when trying to query when a bone is in motion, rather than the severity of motion.
Component Stripping	Here you can select the specific axis of motion to strip from the queried data. Selecting None performs no component stripping, selecting Strip XY will only observe the X and Y axis of motion, matching along the horizontal plane, and selecting Strip Z will only observe motion on the Z axis, only matching along the height motion of the query.

Experimental Channels

Here you can reference a list of the Experimental Pose Search Schema channels and a description of their functionality:

The following channels are experimental, and their functionality should not be relied upon in production.

Crashing Legs (Experimental)

The **Crashing Legs** channel is used to prevent overlap between a character's legs when in motion by selecting poses that keeps the limbs at a set distance apart. This is calculated using the angle between the direction from the `LeftThigh` position to the `RightThigh` position, and the direction from the `LeftFoot` position to the `RightFoot` position, which is then divided it by PI to have values within the set range of `-1.0` to `1.0`. This channel is automatically injected into the Pose Search Schema, when the Number of Permutations property is greater than `1.0`. This channel is designed to be used in conjunction with the experimental **Motion Matched Stitching** feature.

Here you can reference a list of the Crashing Legs Channel’s settings and a description of their functionality:

Property	Description
LeftThigh	Set the bone that represents the character’s left thigh.
Right Thigh	Set the bone that represents the character’s right thigh.
Left Foot	Set the bone that represents the character’s left foot.
Right Foot	Set the bone that represents the character’s right foot.
Weight	Set a weight value to weigh this channel’s results against the other active channels in the Motion Database Config asset.
Input Query Pose	Here you can set the method used to input the queried pose. You can select from the following methods:

Property	Description
	<ul style="list-style-type: none"> • Use Character Pose: Uses the character's current pose to input the queried pose. • Use Continuing Pose: Uses the character's continuing pose to input the queried pose. • Use Interpolated Continuing Pose: Uses an interpolation blend to input the queried pose with the continuing pose.
Allowed Tolerance	Set the allowed tolerance when making selections using this channel.

Permutation Time (Experimental)

Using the Permutation Time channel you can weight additional channels in order to set the amount of influence each channel has over the final output pose.

You can set the weight of this channel's influence compared to the other channels using the **Weight** property.

Create a Pose Search Database Asset

In order to store your character's animation data in a format the Motion Matching node can make selections from, you must create a Pose Search Database asset. To create a Motion Database asset, navigate in the **Content Browser** to **(+) Add** and select **Animation > Motion Matching > Pose Search Database**.

Q Start typing to search

GET CONTENT

- Import to /Game...
- Add Feature or Content Pack...
- Add Quixel Content

FOLDER

- New Folder

CREATE BASIC ASSET

- Blueprint Class
- Level
- Material
- Niagara System

CREATE ADVANCED ASSET

Animation

Artificial Intelligence

Audio

Blueprint

Cinematics

Editor Utilities

Foliage

FX

Gameplay

Input

Interchange

Material

Media

Miscellaneous

Paper2D

Physics

Texture

Tool Presets

User Interface

World

Q Start typing to search

- Aim Offset
- Animation Blueprint
- Animation Composite
- Animation Layer Interface
- Animation Montage
- Blend Space
- Contextual Anim Scene
- Mirror Data Table
- Pose Asset
- Variable Frame Stripping Settings

Advanced

Control Rig

Legacy

Motion Matching

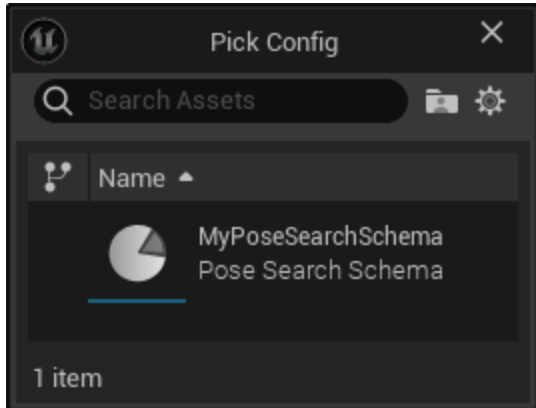
Retargeting

Pose Search Database

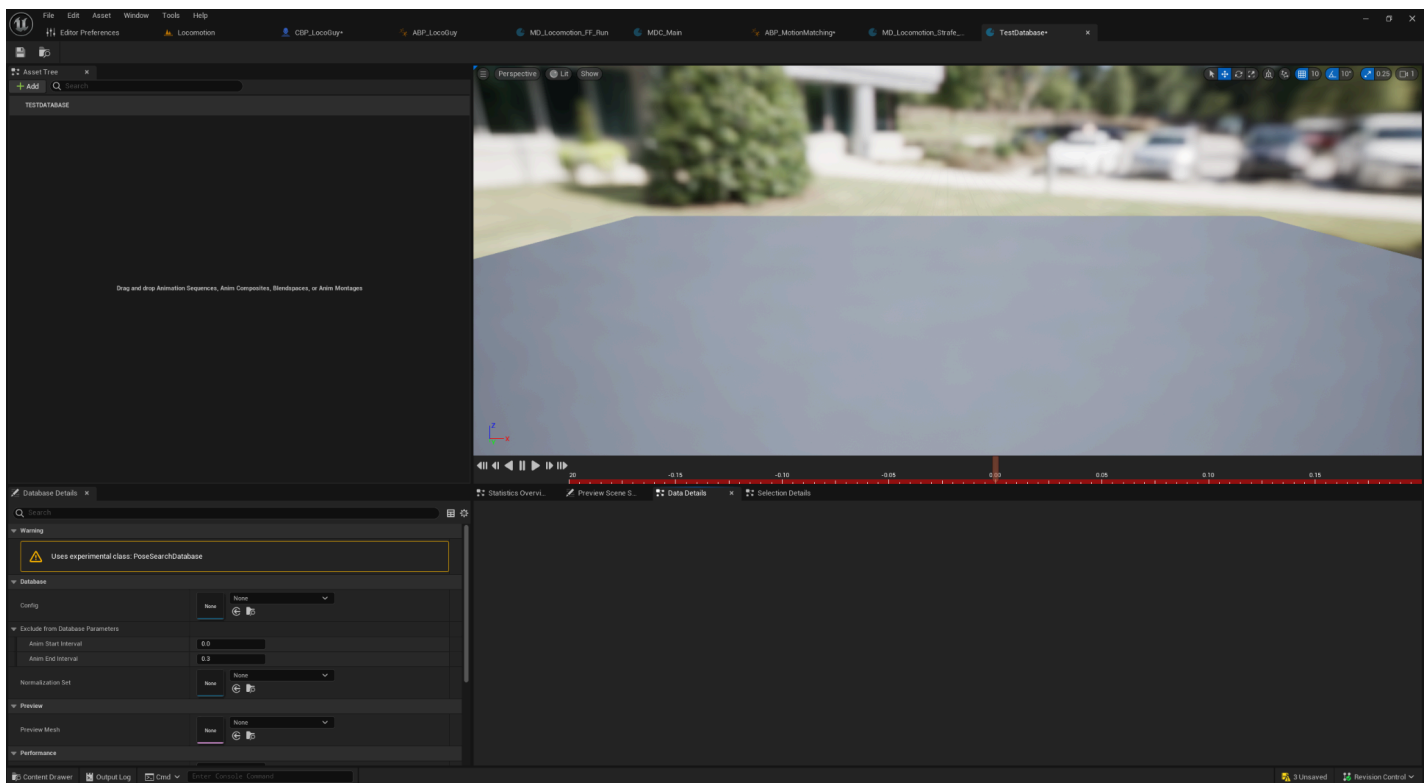
Pose Search Normalization Set

Pose Search Schema

After creating the Pose Search Database asset you must define a Pose Search Schema asset by selecting your Pose Search Schema asset from the asset picker. you can open the asset editor by double clicking it in the Content Browser.



Pose Search Databases are where you can add and edit the list animation assets available to the Motion Matching in order to make selections from. After creating the asset you can open the Pose Search Database asset editor by **double-clicking** the asset in the **Content Browser**.



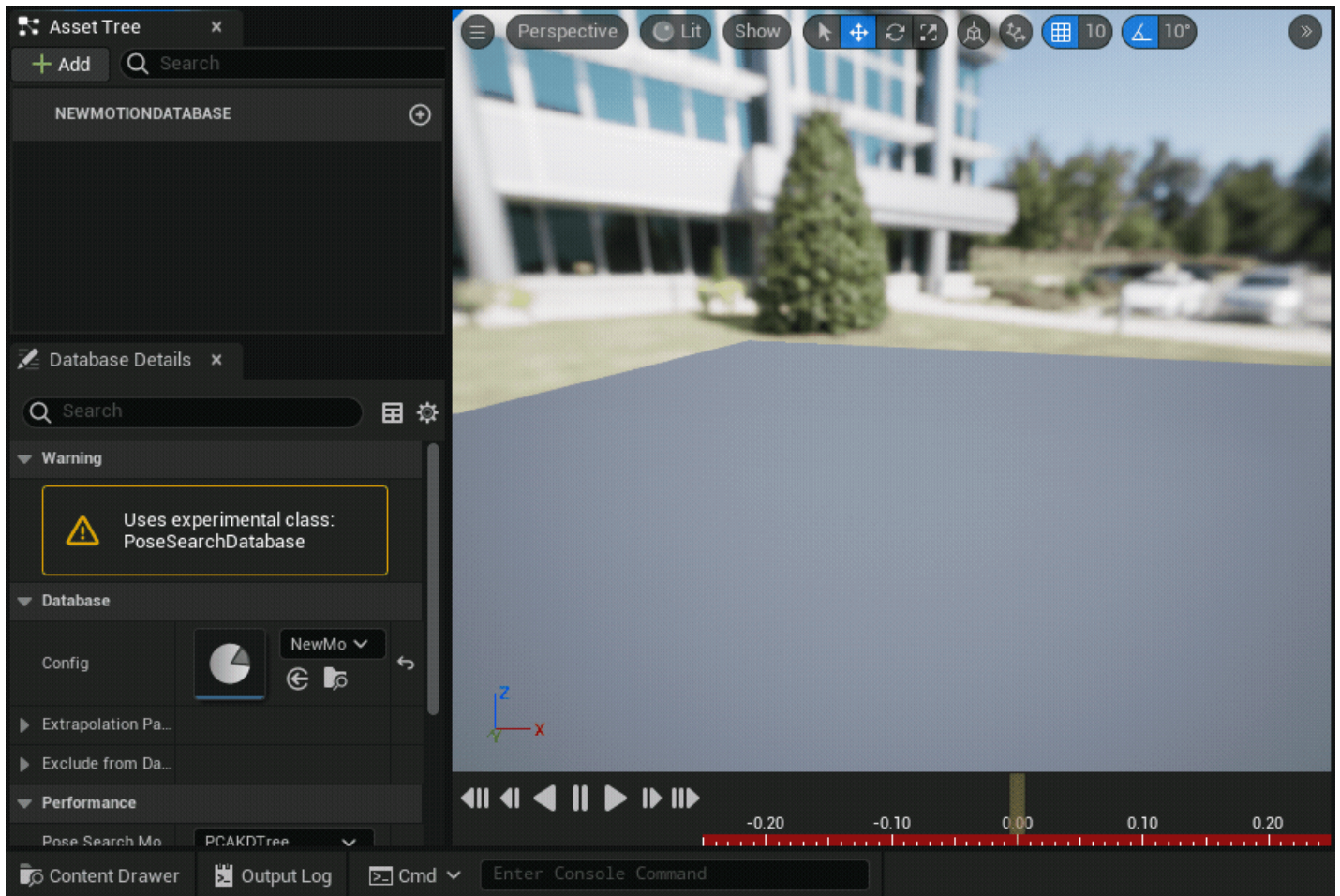
You can now add your animation assets to the Motion Database asset. In this workflow example, a set of locomotion assets, such as loops, pivots, and stops are used.



Animations used in a Motion Matching setup for locomotion must contain root motion, and the Animation Sequences must have the Root Motion property enabled.

You can use any amount of animation assets as you need to achieve your desired fidelity, the more animations assets you provide the more data the Motion Matching node will have to make selections from.

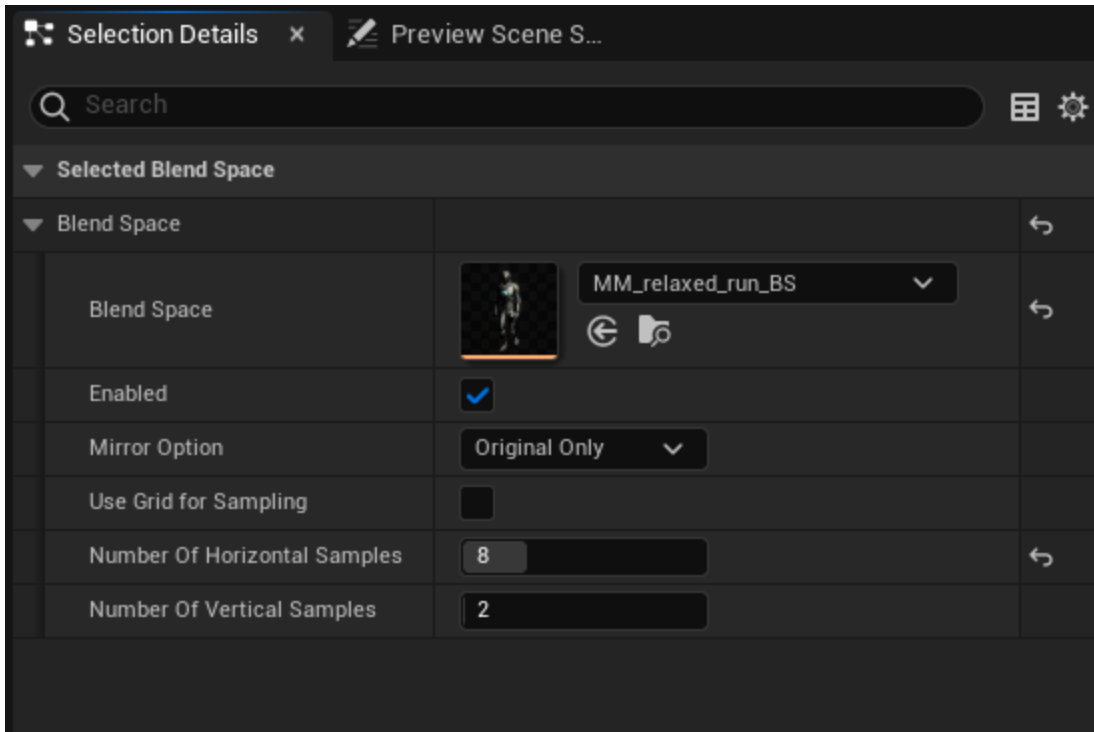
To add an animation asset, you can use (+) **Add** in the **Asset List** panel, or drag and drop assets using the **Content Browser** or **Asset Browser** panels.



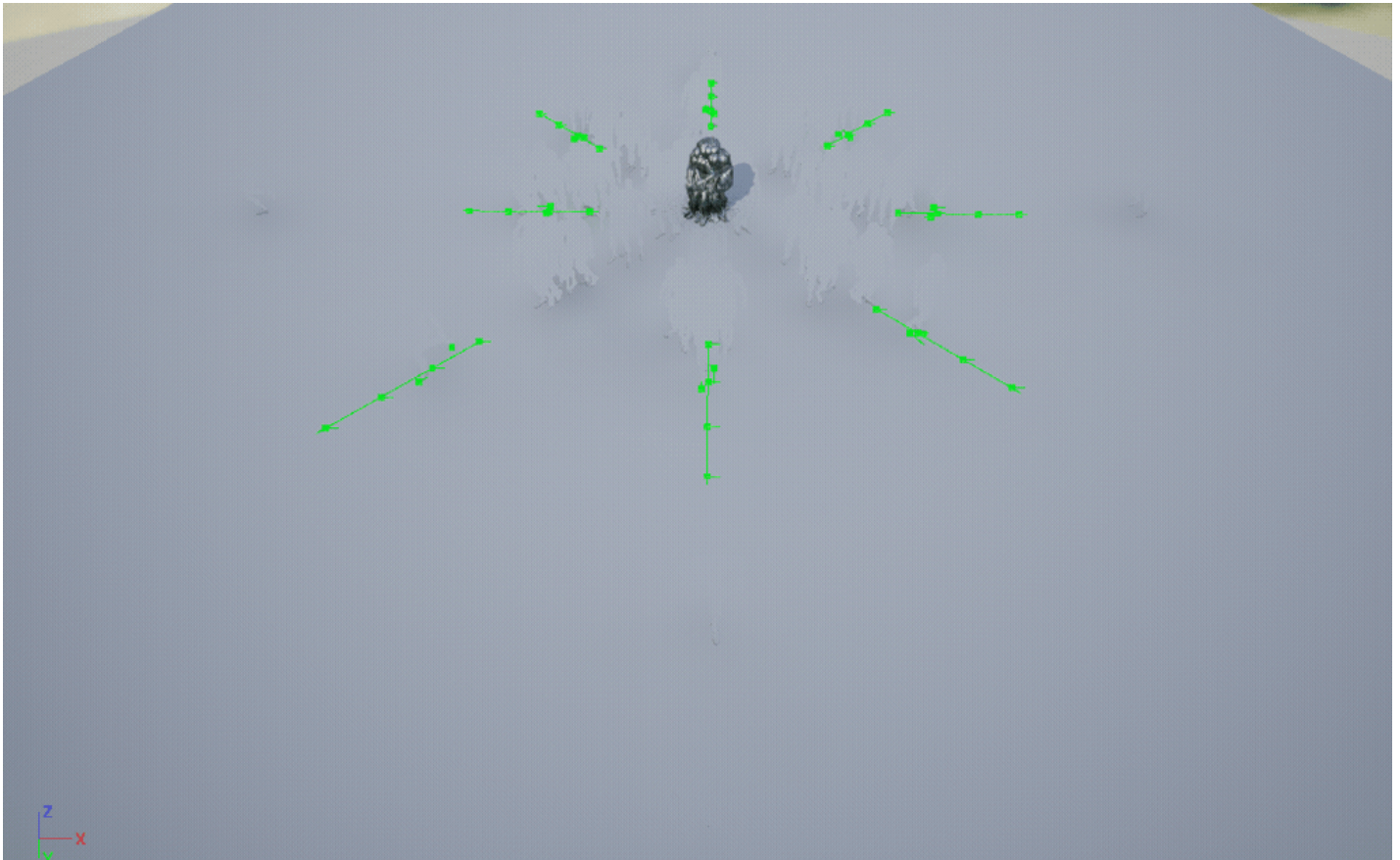
Pose Search Database assets support a variety of animation asset types, while [Animation Sequence](#) assets are the default asset you can use, you can optionally use [Animation](#)



[Composite](#), [Blendspace](#).



While [Animation Montage](#) are not supported in Animation Blueprints, and are therefore not usable with the Motion Matching node, Montage assets can be used and played back in Motion Matching setups using the experimental Motion Matching function in the Character Blueprint.



After adding animations to the Pose Search Database, you can select one, or multiple Animation Sequence assets in the **Asset List** panel to view their playback in the **Viewport** or edit their properties in the **Selection Details** panel.

Statistics Overvi... x

Preview Scene S... x


Data Details x

Selection Details x



Search

Selected Sequence

Sequence



MM_relaxed_run_pivot_BR_FL_footL v



Sampling Range

Min 0.0 Max 0.0

Mirror Option

Original Only v

Enabled

☒

Looping


☐

Has Root Motion



☒


Database

Schema



PSS_Default v





Continuing Pose Cost Bias

-0.01

Base Cost Bias

0.0

Looping Cost Bias

-0.005

Exclude from Database Parameters

Min 0.0 Max -0.3


Additional Extrapolation Time

Min -100.0 Max 100.0



Tags


0 Array element + -

Normalization Set



PSN_Dense_All v







Preview

Preview Mesh

None

None v



Performance

Pose Search Mode

PCAkdTree v

Number Of Principal Components

4

KDTree Max Leaf Size

16

KNNQueryNumNeighbors

200

Pose Pruning Similarity Threshold



0.0



PCAValues Pruning Similarity Threshold

0.0

Property	Description
Schema	Here you can set or reference the Pose Search Schema asset, that sets the channels this Pose Search Database asset will use to match against. This property will automatically populate with the Schema you select when creating the Database asset. You can change the associated Schema using the property's asset selection drop down menu.
Continuing Pose Cost Bias	Here you can set a value that will add a bias to the currently selected pose. This allows you to apply a positive or negative cost bias to the pose to influence how it will be valued when selecting the next pose. This is useful to help the system hold on one animation segment longer, or shorter, depending on the value you have set. Negative values make the animation more likely to be continually selected, or played for a longer duration, while positive values make it less likely to be selected again, resulting in the animations being played for a shorter duration.
Looping Cost Bias	Here you can set a value that will be added to all looping animations assets in the database. This allows you to control how likely the system will select a looping animation. Negative values make it more likely that the system will select looping animations, and remain in the looping animations, while positive values make it less likely that looping animations will be selected or played consecutively.
Exclude from Database Parameters	Here you can set how far beyond the first and last frame in an Animation should be evaluated. This property can be used to trim starting and ending frames of the animations in the database to preserve transition poses while blending between the current and next selected animation, such as preventing the very last frames of the animation when blending out. A value of <input type="text" value="0.0"/> will not perform any trimming, while negative frames will trim the animation on either end depending on the property used.

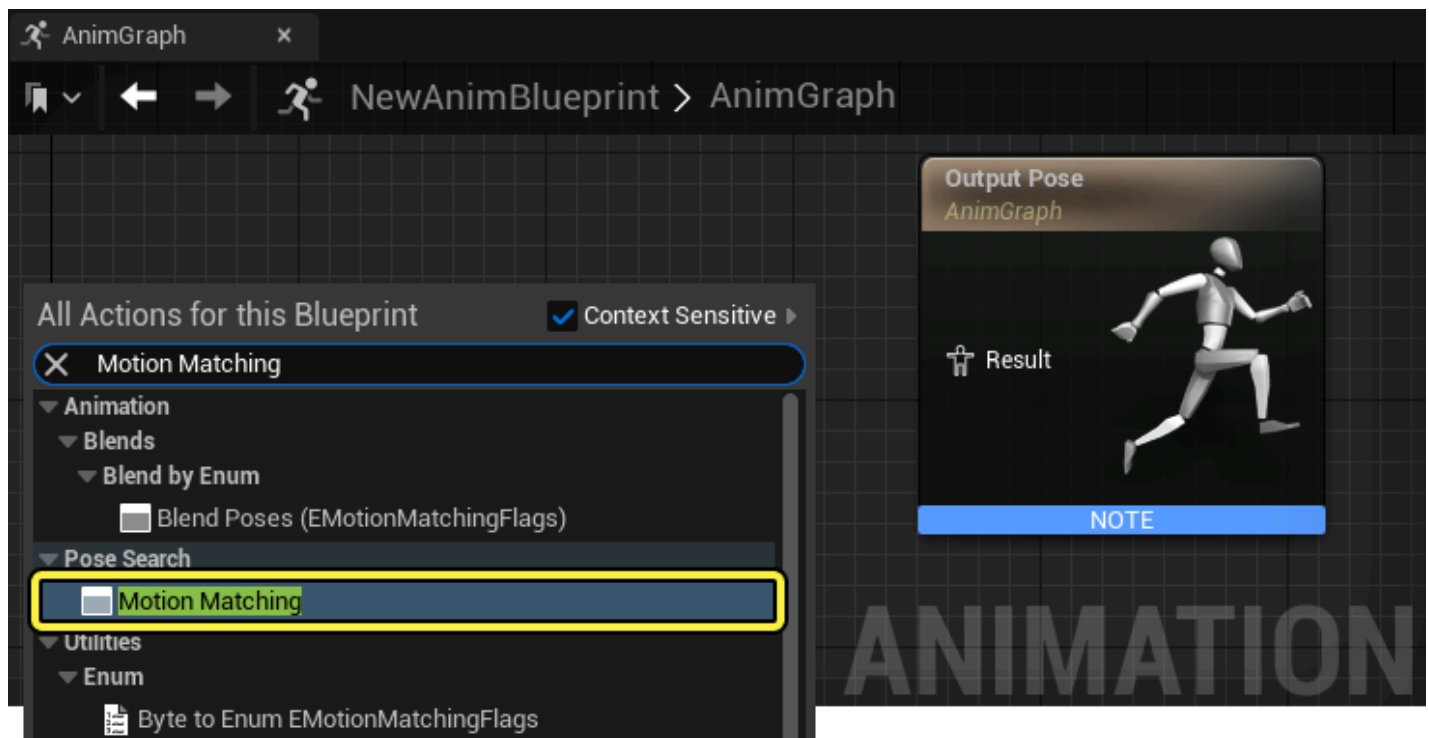
Property	Description
Additional Extrapolation Time	<p>Here you can set values that will determine any additional extrapolation time that will be used when blending the animation in or out. You can set a minimum or maximum value that will be used in the following formulas that will be used to clamp each animation's extrapolation time:</p> $\text{MinimumExtrapolationTime} = \text{AnimationAssetTimeStart} + \text{AdditionalExtrapolationTime.Min}$ $\text{MaximumExtrapolationTime} = \text{AnimationAssetTimeEnd} + \text{AdditionalExtrapolationTime.Max}$
Tags	<p>Here you can set an array of tags that can be used as metadata for the Pose Search Database asset. You can add a new element to the array using (+) Add. After adding an array element you will be able to add a text value to the index element.</p>
Normalization Set	<p>Here you can set a Pose Search Normalization Set asset that will define a list of Pose Search Database assets you want to normalize selection with. While setting a Normalization Set is not required for a functional Motion Matching setup, splitting animation sets up by type and using a Normalization Set to compare sets against each other can yield more accurate and efficient results. For more information about using Normalization Sets, see the Pose Search Normalization Sets section of this document.</p>
Preview Mesh	<p>Here you can set a Skeletal Mesh asset that will be used in the viewport when playing back animations. If left undefined, the editor will use the Skeletal Mesh associated with the animation assets you have added. You can select a new preview Skeletal Mesh, using the property's asset selection drop down menu, and selecting any skeleton that is compatible with the Database's animation assets.</p>
Pose Search Mode	<p>Set the mode in which the database will perform searches of the animation data. Using the drop down menu you can select from the following options:</p>

Property	Description
	<ul style="list-style-type: none"> • Brute Force: Database searches will be evaluated extensively. The system will evaluate all the indexed poses to search for the best one. • PCAKDTree: This is the optimized search mode. The database projects its poses into a PCA (Principal Component Analysis) space using only the most significant dimension derived from the **Number Of Principal Components property, and constructs a KDTree **(a space-partitioning data structure) to facilitate the search. • VPTree: This is an optimized search mode that uses a vantage point tree. <div data-bbox="695 747 1481 930">  <p>This mode is an experimental mode and therefore its functionality should not be relied upon in a production.</p> </div>
Pose Pruning Similarity Threshold	<p>Here you can set a value to determine the threshold for saving similar poses under the same index. For example, if two poses have the same selection values, such as when the multi-dimensional point with the schema cardinality are similar, and these values are closer than the value set for the Pose Pruning Similarity Threshold property, only one will be saved into the database and accessed by the two different pose indexes. This property is useful to set thresholds where multiple poses can be selected for a given value, in order to save memory.</p> <div data-bbox="695 1522 1481 1705">  <p>This Property is only accessible when the Pose Search Mode property is set to PCAKDTree.</p> </div>
Number Of Principal Components	<p>Set the number of dimensions used to create the KDTree, when the Pose Search Mode property is set to PCAKDTree. Higher values will result in more dimensions, which allows for a better explanation of the variance of the</p>

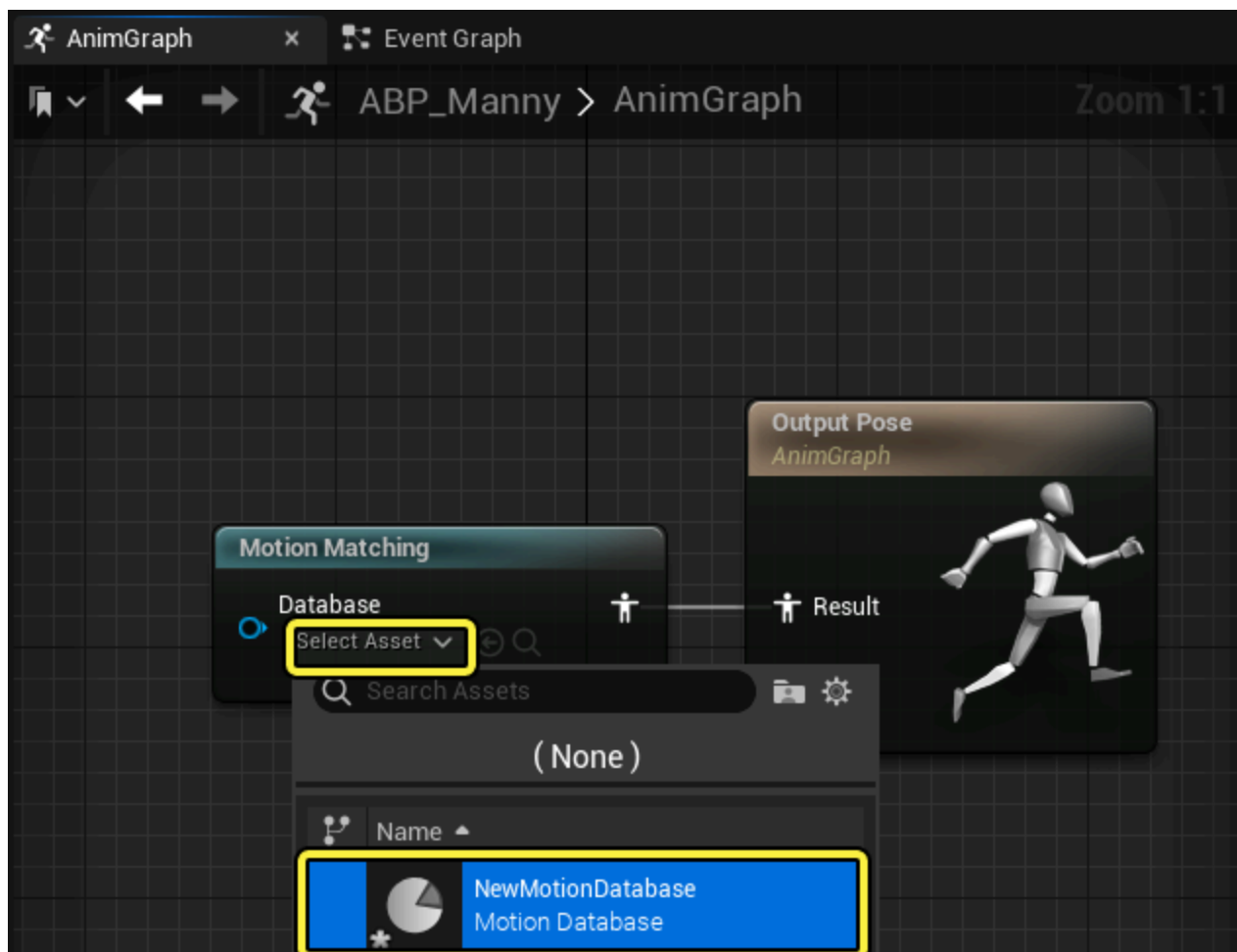
Property	Description
	<p>dataset. More dimensions usually results in better search results, but will also result in more memory usage and worse performance for your project.</p> <div>  <p>This Property is only accessible when the Pose Search Mode property is set to PCAKDTree.</p> </div>
KDTree Max Leaf Size	<p>Set the maximum number of leaves each branch in the KDTree will have.</p> <div>  <p>This Property is only accessible when the Pose Search Mode property is set to PCAKDTree.</p> </div>
KNNQueryNumNeighbors	<p>When using a KDTree search, results will have only an approximate cost, so the database search will select a few poses to perform the full cost analysis, and be able to select the best pose. Set a value in this property to set the number of poses that will be selected for a full cost analysis from the KDTree search. This property can also be used to set the number of poses from a VPTree search, to perform a full cost analysis of, when using the VPTree Pose Search Mode.</p>

Create the simplest setup: Motion Matching Node and Pose History Node

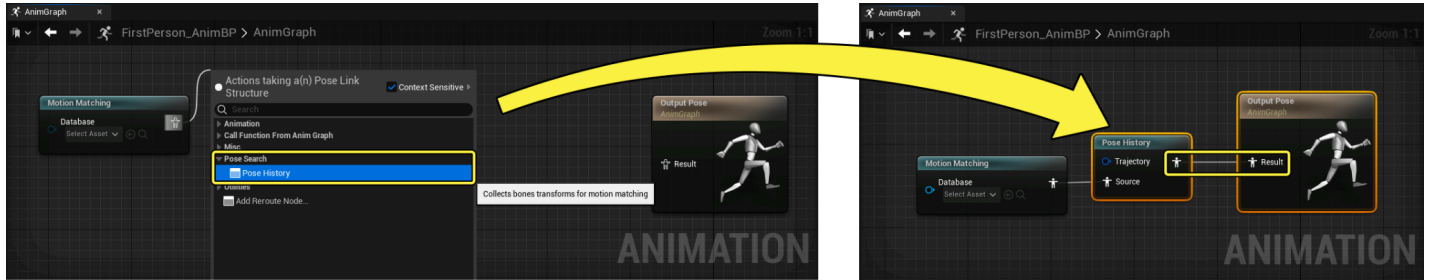
In your character's Animation Blueprint, create a **Motion Matching** node, by right-clicking in the graph and selecting Motion Matching under the **Pose Search** category.



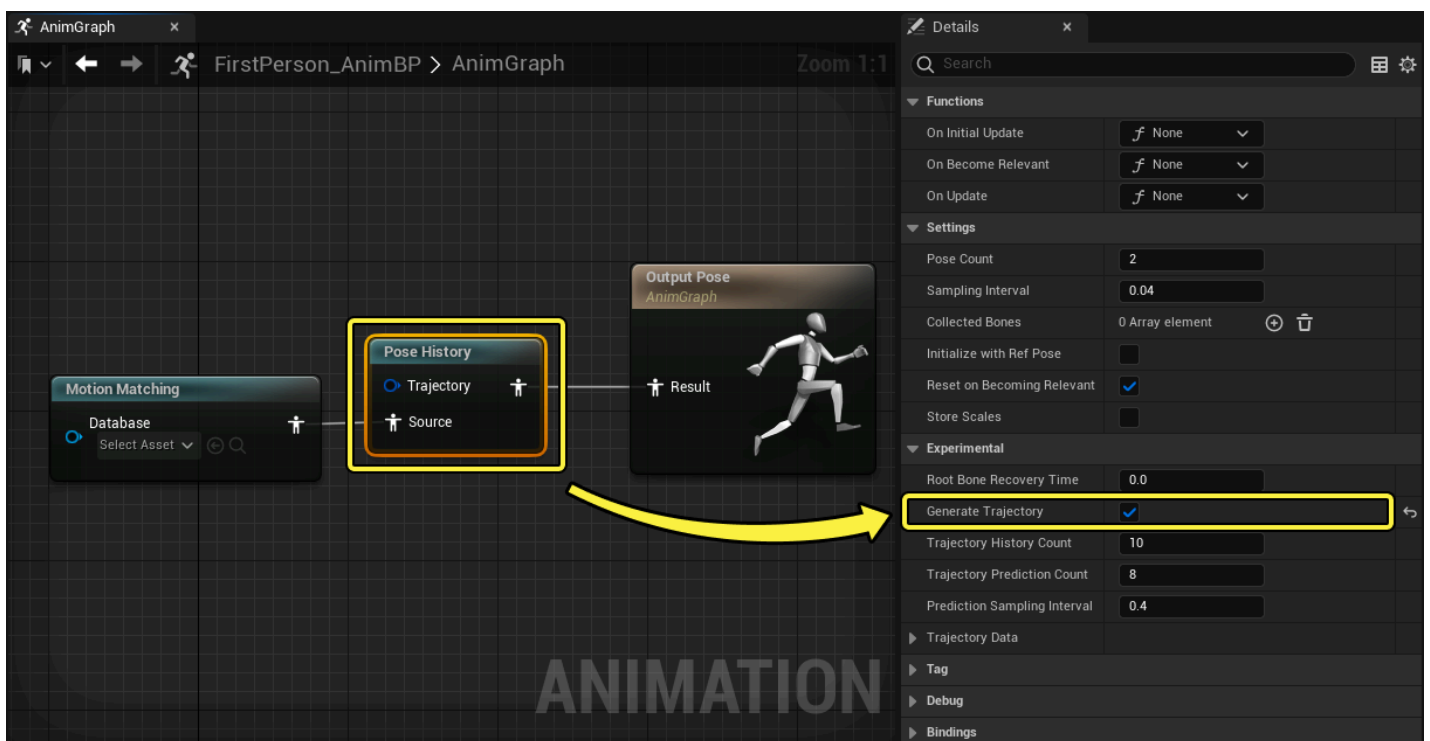
Select your Database asset using the Motion Matching node's **Searchable** pin drop-down menu in the graph.



Next add a **Pose History** Node, by dragging off the Motion Matching node's **Output** pin, and selecting the Pose History node under the **Pose Search** section, or by using the **Search Bar**. Then connect the Pose History node's **Output** pin to the **Output Pose** node.

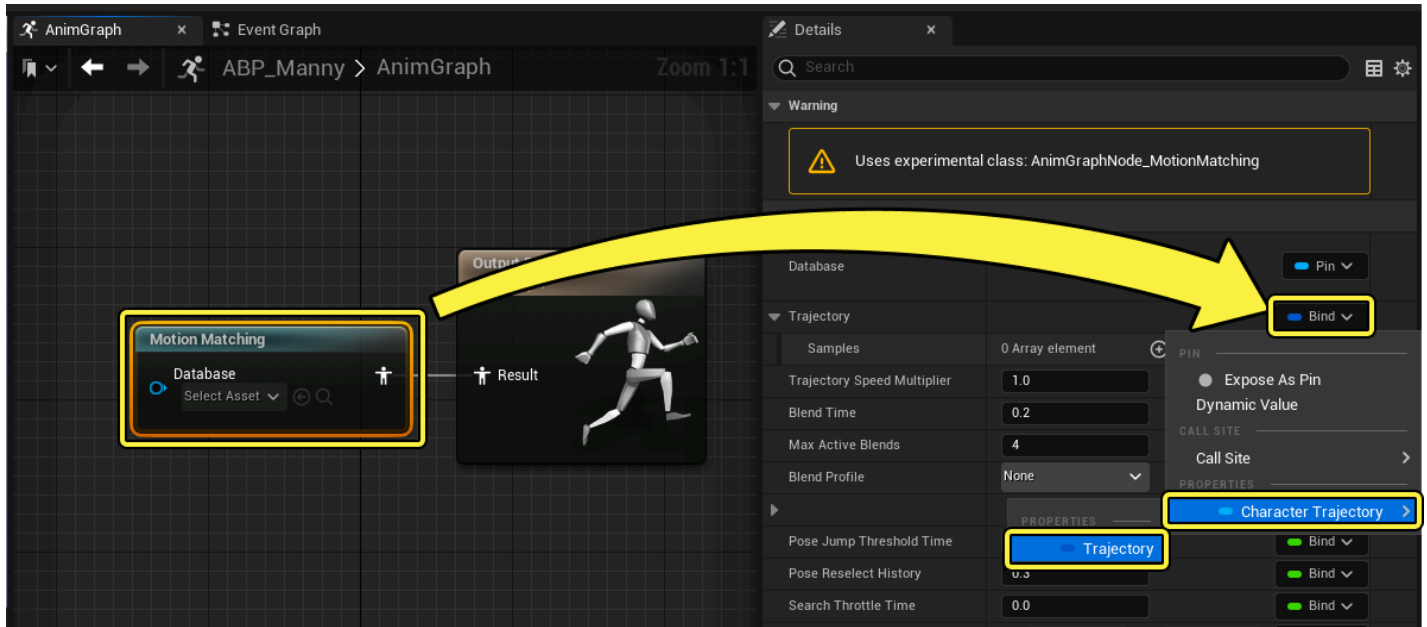


Select the Pose History node in the AnimGraph to expand its properties in the **Details** panel. Enable the **Generate Trajectory** property.



Then, expand the **Collected Bones** property and add an array element using (+) **Add** for each bone in your character's skeleton that you would like to sample. The Bones defined in this property will be sampled when the Pose History and Motion Matching nodes query the

Trajectory when making animation selections.

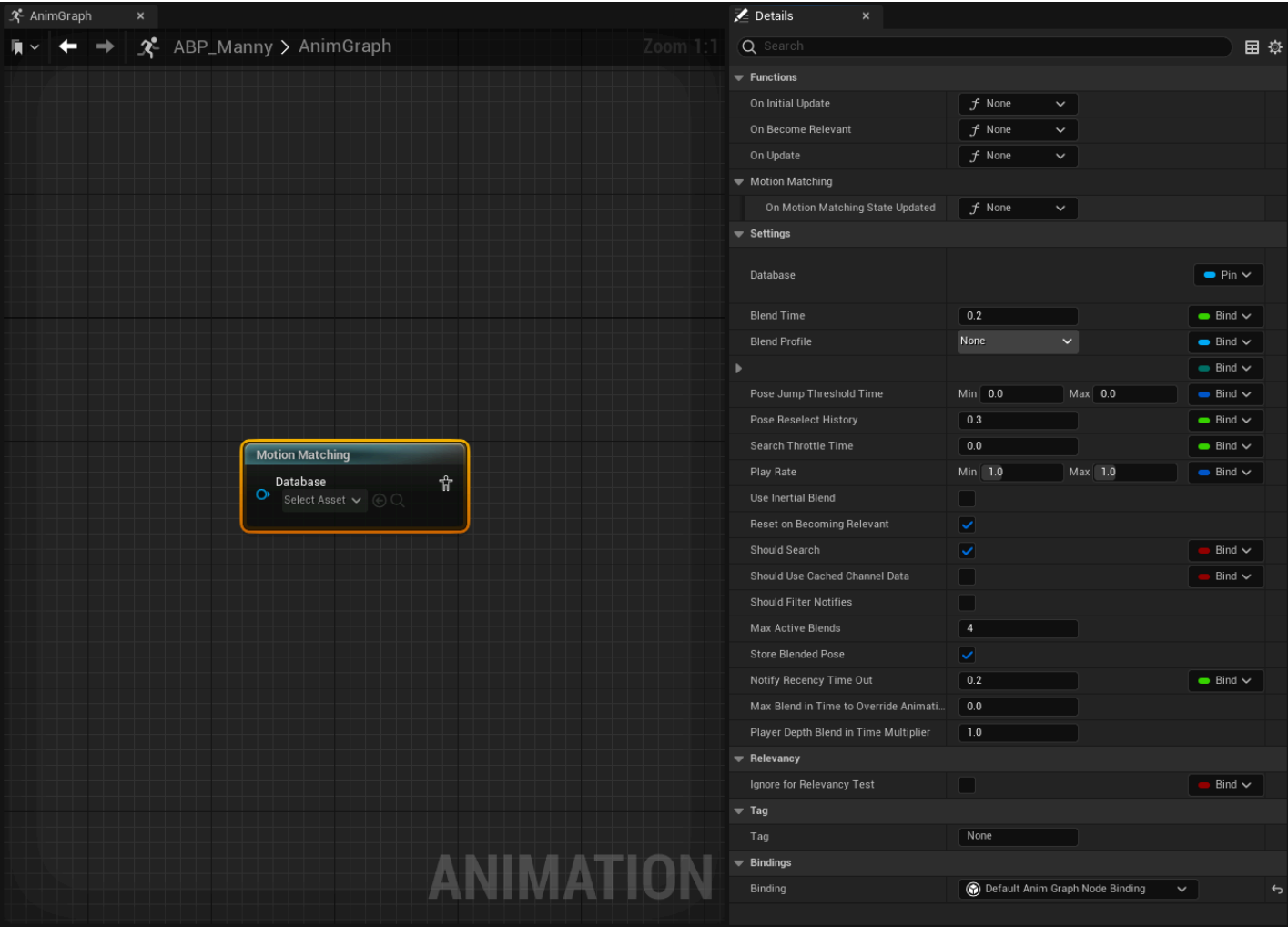


Save and **compile** the Animation Blueprint. Your character is now using Motion Matching to select a locomotion animation pose at runtime.



Motion Matching Node

Here you can reference a list of the Motion Matching node's settings and a description of their functionality:



Property	Description
Database	Here you can set your Database asset, that will be used to make selections from.
Blend Time	Set a value that will be used to determine how long the blend will be when selecting a new animation sequence segment. Motion Matching has a built-in Blend Tree that allows us to have more options for Blend types, quality of Blends, and Blend Profiles. You can do things like Blend the upper body longer than the lower body for example, if you wanted to. This will be expanded out with even more options in the future.

Property	Description
Blend Profile	Here you can set one of your character's Blend Profiles to be used with Motion Matching.
Mode	Here you can set a curve that will be used to determine how the blend is applied over time to the bones. Common selections are Linear , Ease In , Ease Out , and Ease In and Out . You can access this property using the drop down menu nested under the Blend Profile property expansion.
Pose Jump Threshold Time	Here you can set a timeframe, in seconds, where the system will not jump to poses within this range, on the same asset, unless it's this amount of time away from the current frame. This helps prevent the system from picking sections of animation inside the same clip over and over and helps encourage playing out animations or picking something new.
Pose Reselect History	Here you can set a range, in seconds, that prevent the system from jumping to poses that have been selected previously within the amount of seconds set, in the past.
Search Throttle Time	Set how often the system searches for new poses.
Play Rate	Here you can set the play rate for the animation selections.
Use Inertial Blend	When enabled an Inertial Blend will be applied to animation blends.
Reset On Becoming Relevant	When enabled, this property will reset the Motion Matching state if the node has become relevant to the graph after not being ticked on the previous frame or frames.

Property	Description
Should Search	When enabled the Motion Matching node will search the connected Database asset for animations to match against the query system.
Should Use Cached Channel Data	When enabled, the search of multiple databases with different schemas will try to share pose features data calculated during the query build. The intention is to be able to share as much as possible the continuing pose features vector across different schemas (and potentially improve performances). By default this property is disabled to preserve backwards compatibility with existing motion matching systems.
Should Filter Notifies	When enabled, any notifies from originating from an anim player samples should be filtered. {in seconds}
Max Active Blends	Set the maximum number of active blending animations in the blend stack. If this property is <input type="text" value="0"/> then the blend stack will be disabled.
Store Blended Pose	When enabled, and the number of requested blends is higher than the value set for the Max Active Blends property, the blend stack will blend and accumulate into a stored pose with all the overflowing animations. When disabled, the memory to store the pose will be saved, but once the value of the Max Active Blends property is reached, the blendstack will start discarding animations, potentially resulting in animation popping.
Notify Recency Time Out	Set the window of time, after firing a notify, that any instance of the same notify will be filtered out.
Max Blend in Time to Override Animation	If the most recently added animation is within the range set by this property's value, the new requested blend will take its spot. Otherwise a new blend will be added to the blend stack.

Property	Description
Player Depth Blend Time Multiplier	Set a value that will multiply the Animation Player's blend in timer. Increasing this value can be useful in producing faster blend times on deeper blends.

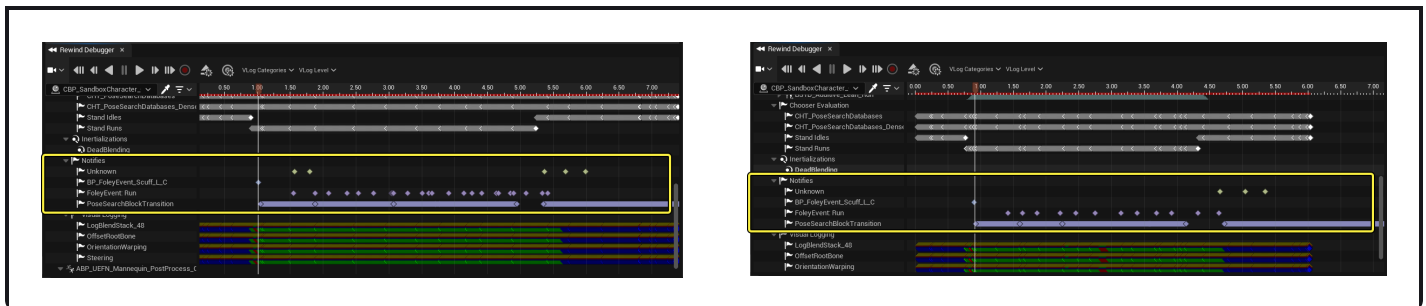
Anim Notify Filtering

When using the Motion Matching system, you can encounter issues where the poses that the system selects, each trigger a specific Anim Notify, such as a footstep sound effect during a walking or running animation. This results in two or more footstep sound effects playing on adjacent frames, rather than just one. This issue can be solved using **Anim Notify Filtering**.

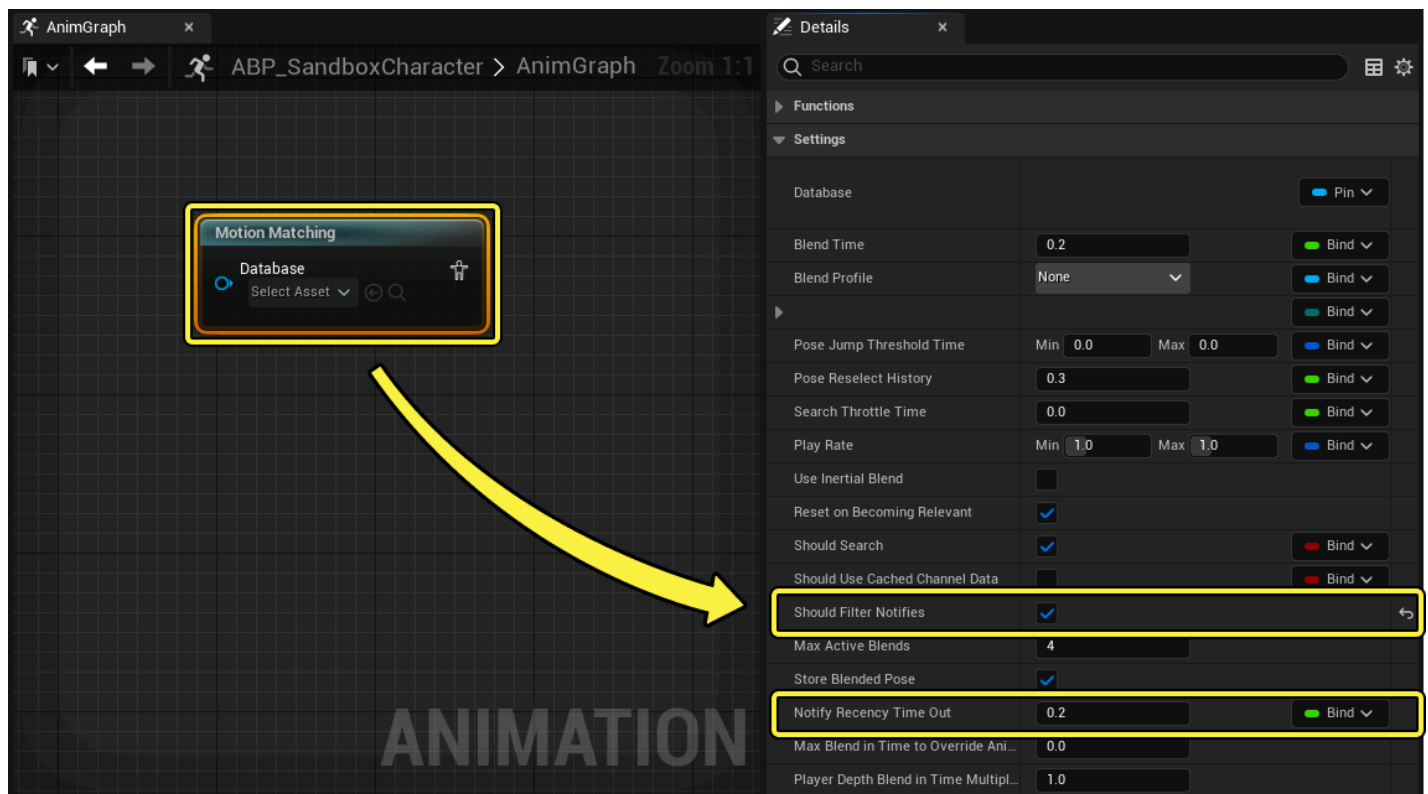
Here you can see two Rewind Debugger recordings of the same character gameplay situation, the example on the left is not using Anim Notify filtering, and the sequence is firing the same Anim Notify more than once due to the Anim Pose selections. The example on the right is using Anim Notify Filtering to prevent the same notify from being triggered within a set time frame, resulting in the Anim Notify to only be triggered once in a set amount of time.

No Anim Notify Filtering

With Anim Notify Filtering

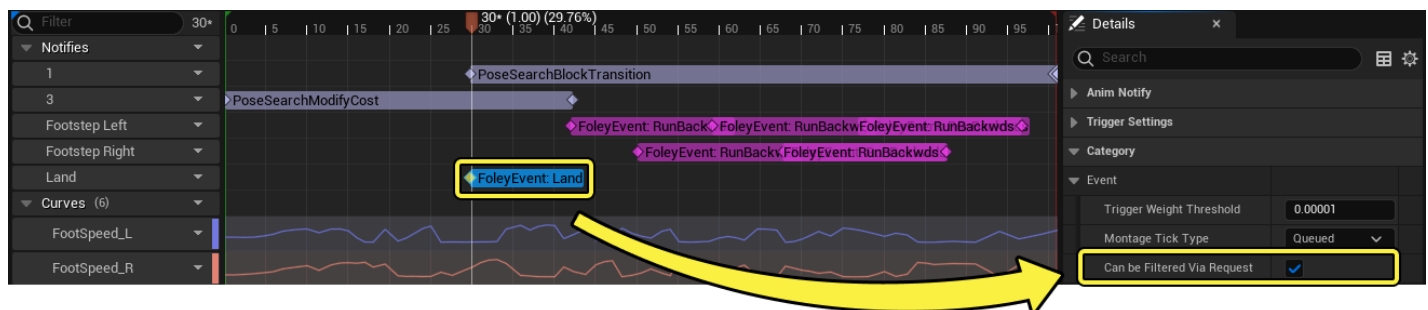


To implement Anim Notify Filtering, select the Motion Matching node in your character's Animation Blueprint, and enable the **Should Filter Notifies** property in the **Details** panel. You can then set the duration of time in seconds a notify will be filtered out using the **Notify Recency Time Out** property.



The default value of **0.2** can be used as a baseline as it was designed to be the most useful for biped locomotion, however your project may require a different value for your intended results.

When enabling the **Should Filter Notices** property, all Anim Notices will be treated the same, filtering out any repeat instances of the same notify for the set value set in the **Notify Recency Time Out** property. If your animations contain a notify that you do not want filtered out, select the Notify in your Animation Sequence or Montage, and disable the **Can Be Filtered Via Request** property in the notify's **Details** panel. By default this property is enabled for all Anim Notices.

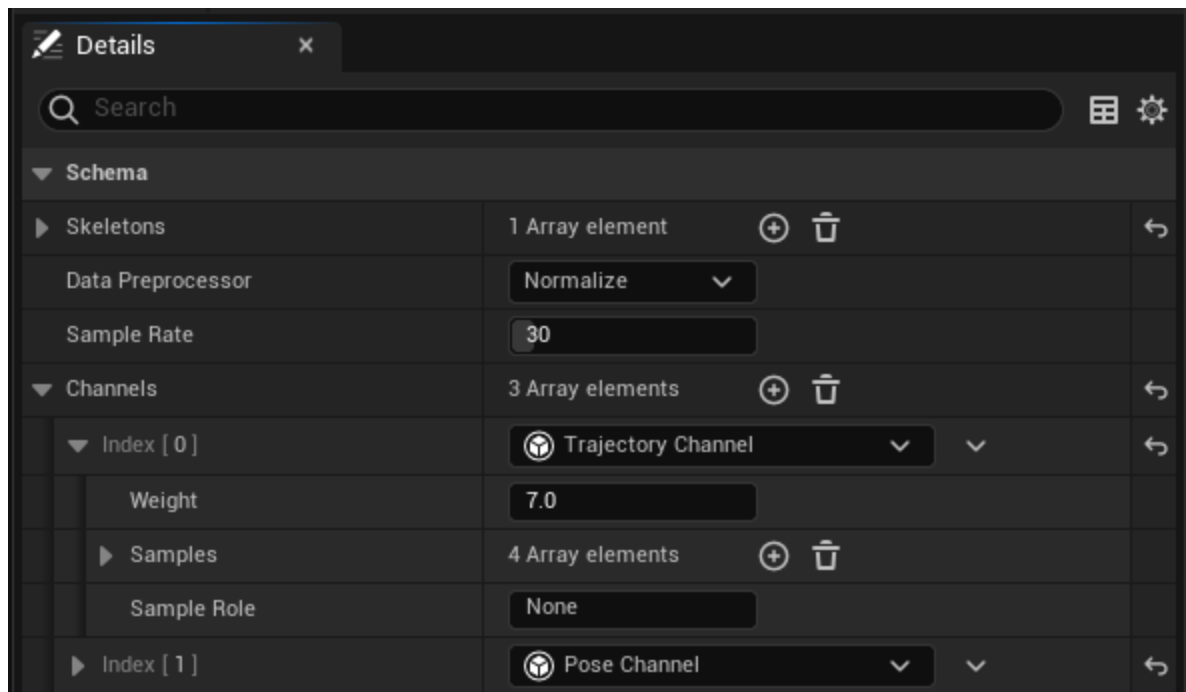




Anim Notify Filtering is only applicable to Anim Notifies, and will not work with Anim Notify States.

Pose Search Schema Channel Weights

You can control how much influence each channel has over the selection process using each channel's Weight property within the Pose Search Schema asset.



The **Weight** property for each channel in the Pose Search Schema asset tells the Motion Matching system how much each channel should influence the selecting of animation frames in the Database.



In the workflow example above, the Pose channel weight set to **1.0** and the Trajectory channel weight set to **3.0**. This means the system will be influenced three times as much about differences in trajectory than it will about differences in the pose.

Individual properties within each Channel can also be given weight to further tune the factors influencing the animation selection. For example, the History and Prediction Horizon properties can also be given weights to bias scoring toward the past or future. This allows

you to distribute what elements of a channel are used to determine its influence amongst other channels in the Pose Search Schema.



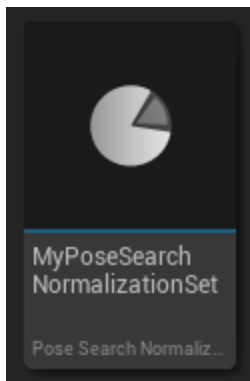
Regardless of what values you give to these various weights, the system will automatically normalize them. To prevent weights from being normalized, set the **Data Processor** property to **None**.

However, weights will not always solve incorrect pose selection. For example, if your movement model is moving at 5 meters a second, but your data is averaged around 4 meters a second, this could result in a Motion Matching system that consistently chooses the maximum value animation poses. Adjusting weight values to devalue the maximum value animation poses may correct this repeated selection but can result in some channel criteria being ignored completely, resulting in gaps in your animation system.

When adjusting the weights of channels you are adjusting how the system is selecting animation poses, rather than adjusting your system to account for your project's systems. In order to preserve your entire animation system, other methods such as using Pose Search Database Sets can be used to account for all gameplay scenarios.

Pose Search Normalization Sets

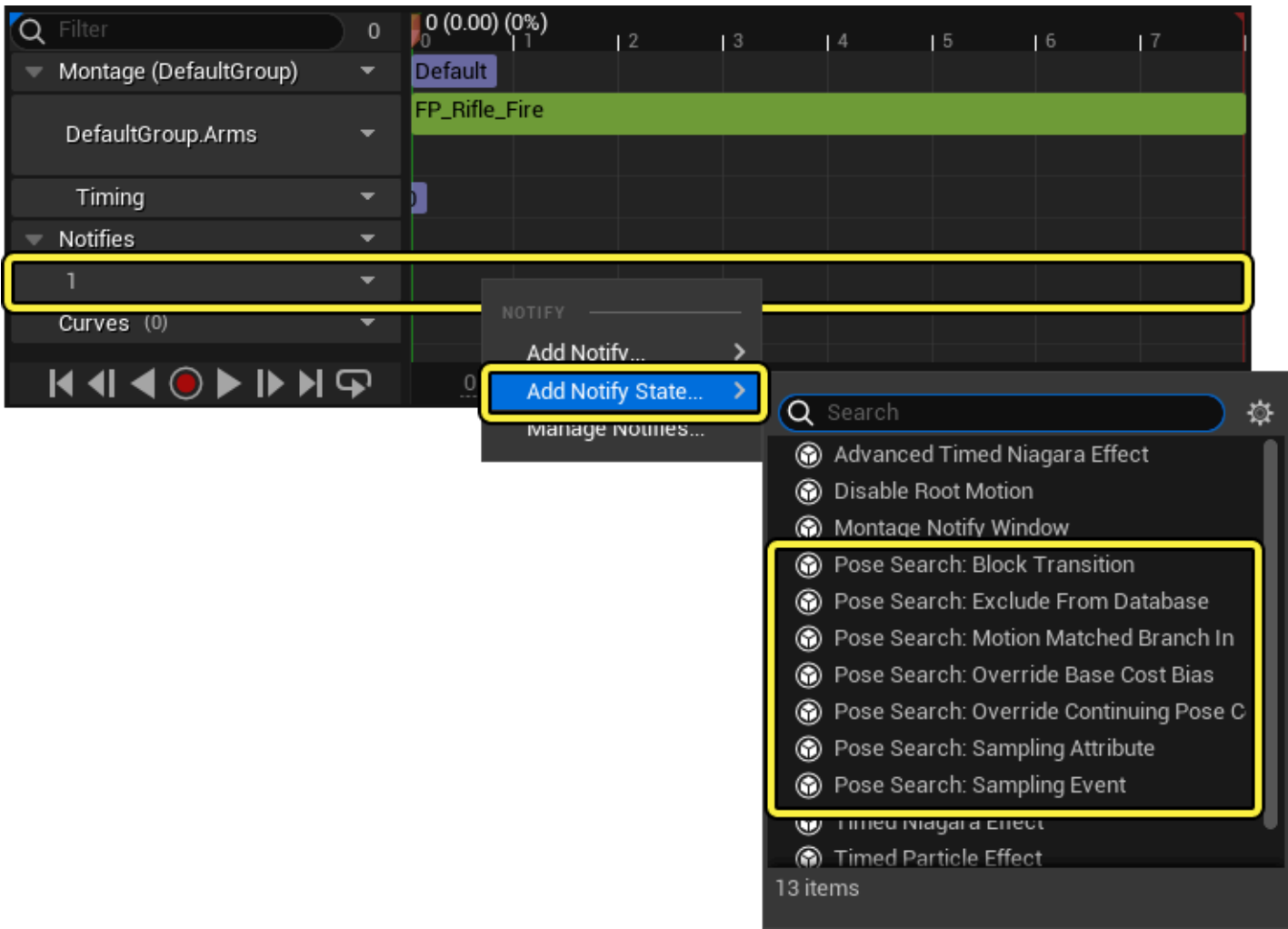
Pose Search Normalization Sets are a series of defined Pose Search Database assets that each contain groups of Animations that are applicable to specific use cases. For example, one database may contain a character's walking animations, while another contains the character's running animations. By setting multiple Pose Search Database assets in a Pose Search Normalization Set asset, you can compartmentalize the animation sets, and set weights for each to become preferable to the Motion Matching system when specific criteria are met.



Pose Search Normalization Sets can be dynamically swapped in when specific gameplay events are triggered. This will limit the Motion Matching node's pool of animation data to select from, giving you more control over the animation pose output of your animation system. This can be done by defining a series of Pose Search Database assets in a Pose Search Normalization Set asset and use a chooser system to dynamically set which Database is active at runtime.

Motion Matching Notifies

You can also use a set of Motion Matching Notifies to assist in your setup. Here you can reference a list of the available Motion Matching notifies, and a description of their functionality:



Notify	Description
Pose Search: Block Transition	Using this Notify State, you can set areas of an animation where the system cannot jump into the

Notify	Description
	<p>tagged area, but it can play into that area from a previous section that is not tagged with this. This is useful when you have sections of an animation you don't want to jump into the middle of. If you have an idle animation that has the character scratching their head in the middle of it, you don't ever want to jump into the middle of the head scratch. If you tag that area you can jump into any sections of the animation not tagged with this.</p>
Pose Search: Exclude From Database	<p>Using this Notify State, you can completely exclude sections of animation data from their Animation Sequences. Any sections set with this notify will not be built or searched against.</p>
Pose Search: Motion Matched Branch In	<p>You can use this Notify State to delineate where an animation should be branched in when being selected by a Motion Matching system.</p>
Pose Search: Override Base Cost Bias	<p>This Notify State allows you to give a section of the animation sequence a bonus or a penalty for selecting this section of the animation. Negative values give it a bonus, while positive values give it a penalty. This is helpful in situations where the system is leaving your start animation earlier than you want, so you could place this notify over the sections you want it to continue on before leaving ideally and give it a small bonus to try and keep it in that area longer.</p>
Pose Search: Override Continuing Pose Cost Bias	<p>Using this Notify State, you can set the Pose Search cost for the continuing pose, making the animation segment more or less likely to be selected again, for continuing playback, based on the notify parameters.</p>
Pose Search: Sampling Attribute	<p>Your Pose Search Schema Channels can use this Notify State as an animation space position,</p>

Notify	Description
	rotation, and linear velocity provider during database indexing by specifying the Channel's Sampling Attribute ID property to match the Notify State's Sampling Attribute ID property.
Pose Search Sampling Effect	Your Pose Search Schema Channels can use this Notify State to demarcate events identified by Sampling Attribute ID property during database indexing by specifying the Channel's Sampling Attribute ID property to match Notify State's Sampling Attribute ID property.

Animation Warping

You can implement Animation Warping nodes in conjunction with Motion Matching to mitigate the lack of coverage by procedurally modifying the character's pose at runtime.

For more information about animation warping, see the following documentation:



Motion Warping

An in depth look at Motion Warping for animations in Unreal Engine.

Motion Matching Debugging

For more information about debugging and optimizing your Motion Matching animation system, see the [Motion Matching Debugging](#) documentation.

More Information

For more information about Motion Matching in Unreal Engine, see the [Motion Matching | GDC 2024](#) presentation.

Sample Content

You can download and explore the Game Animation Sample project, to access a suite of high-quality animation assets and a fully built Motion Matching setup for character locomotion.

For more information about the Motion Matching sample project, see the following documentation:

Game Animation Sample Project

The Game Animation Sample project is an example project that you can use to observe and learn about high quality and responsive animation systems in Unreal Engine.