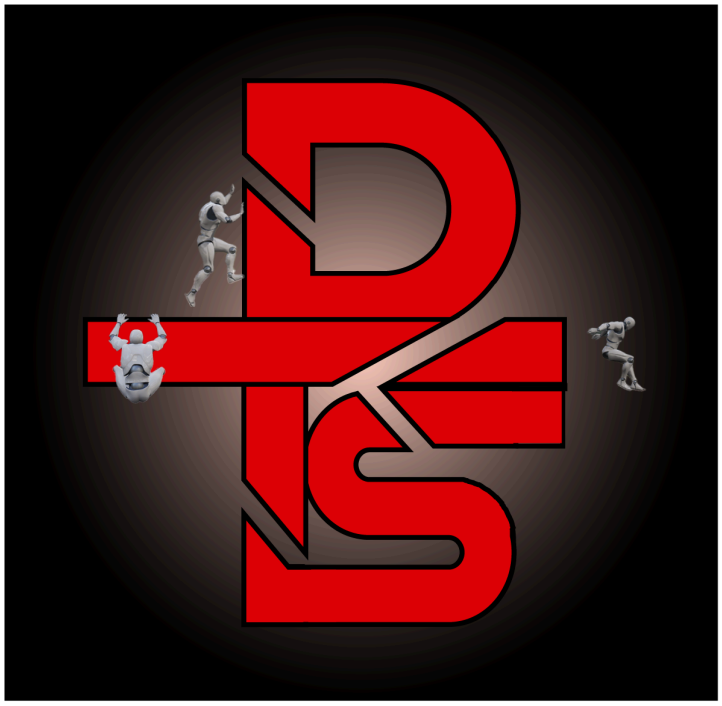
Dynamic Parkour Trace System

Official Documentation



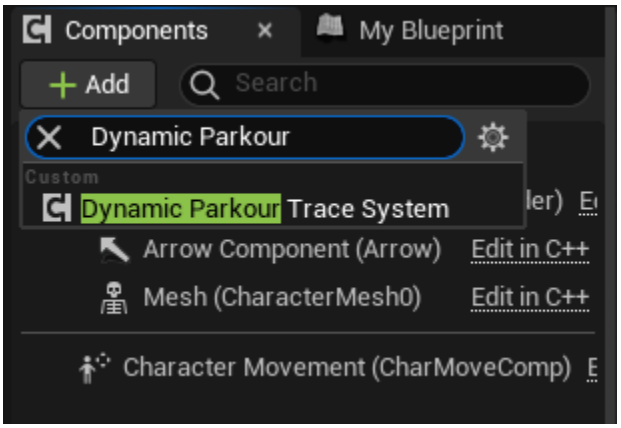
Thank you for Purchasing!!

Hope you Enjoy :)

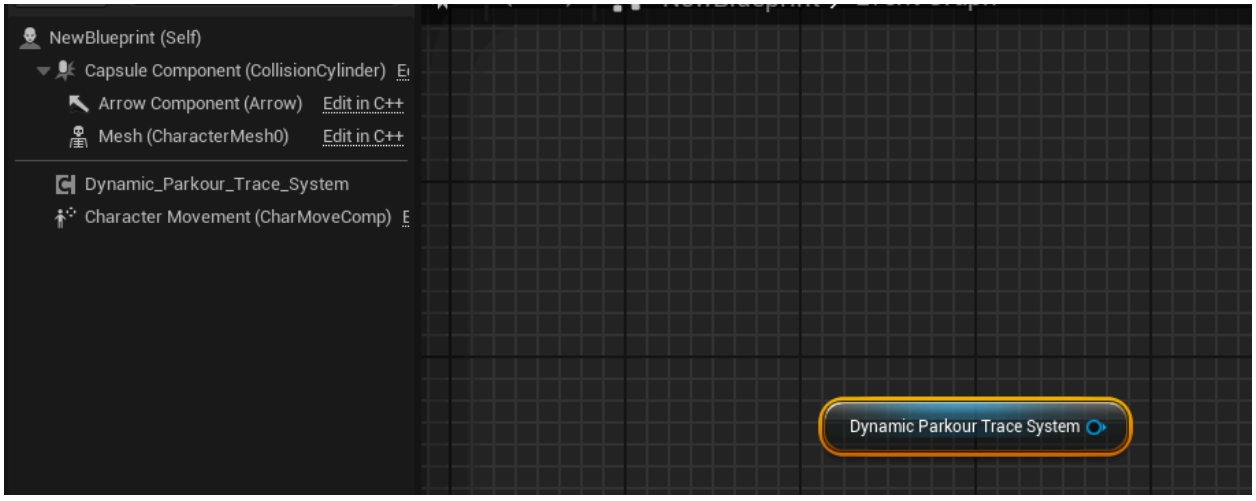
Index

Initial Set up

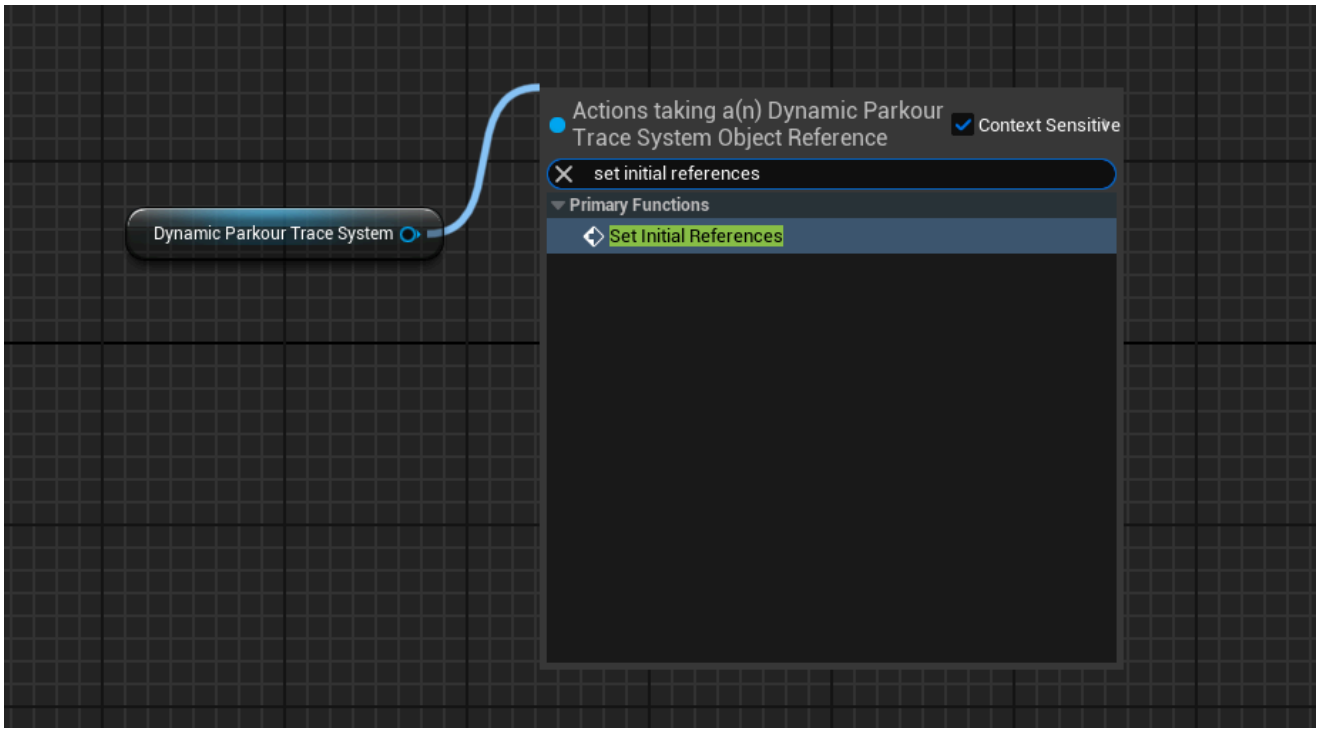
- Open up your character Blueprint, and go to the components section (Defaulted as top left) and click the **+ Add** option and search for **“Dynamic Parkour Trace System”** and add this to your character blueprint



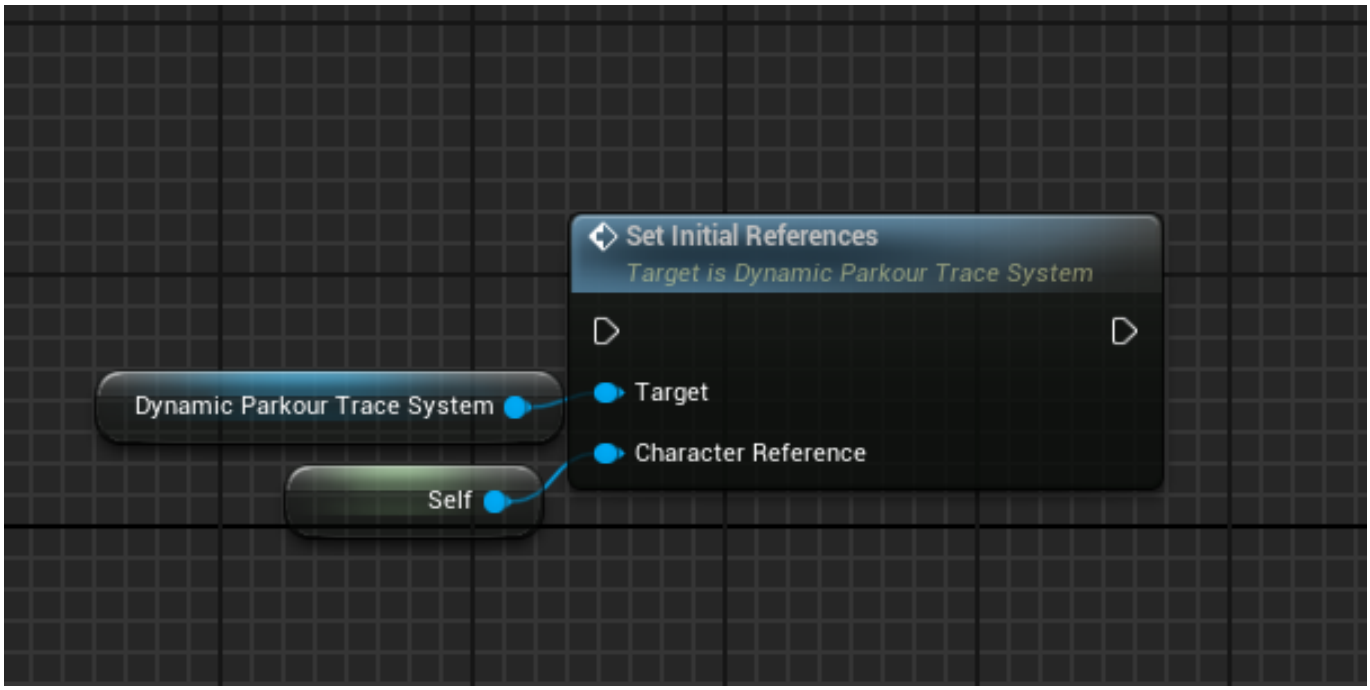
- Click and drag the added component into your graph to acquire a reference



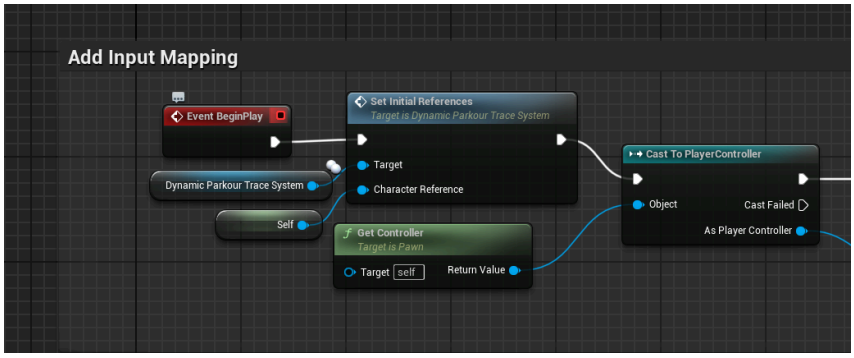
- Drag off the newly created reference and type in **“Set Initial References”**



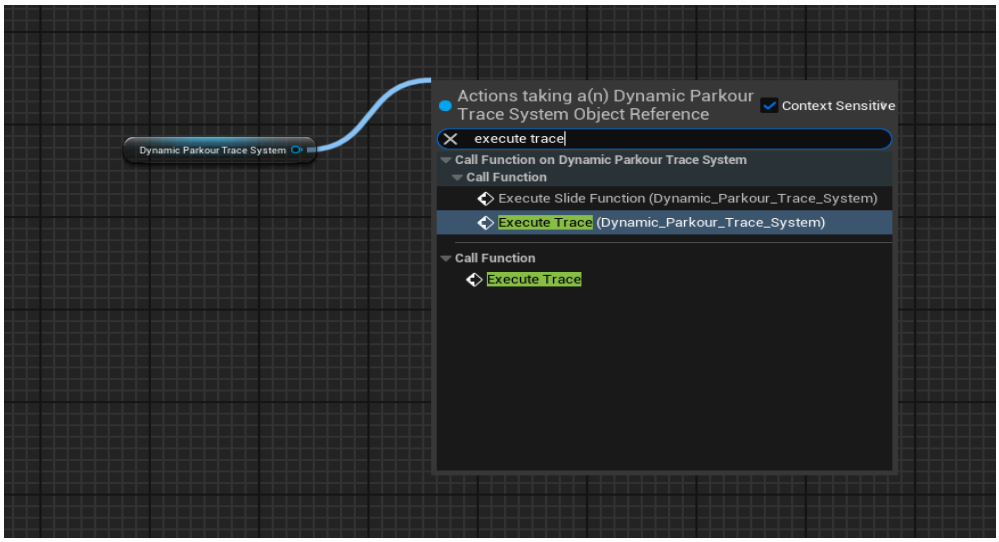
- Pull off the input pin labeled **“Character Reference”** and type in **“Self”** to get a reference to self

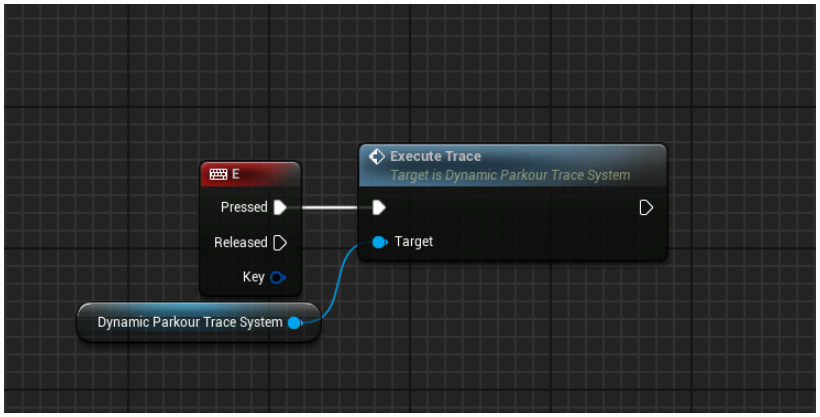


- Add this function to your **“Event Begin Play”** function (image from Third Person Character Template)

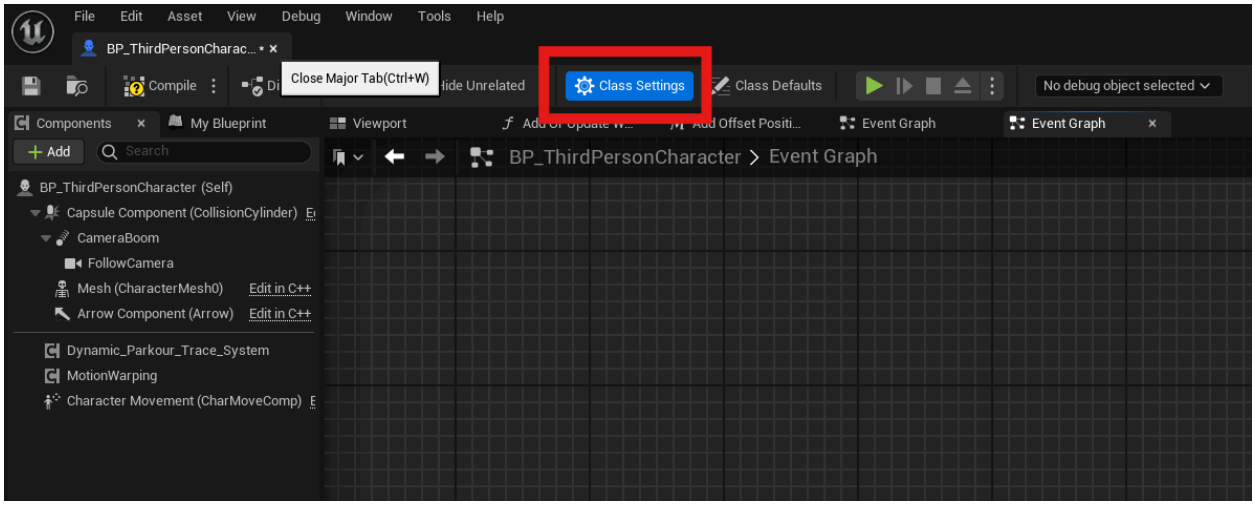


- Once desired input is created (Built in Jump Input can be used - See Modifiers Section for more details) grab a reference to the DPTS System in the same fashion in the previous step, pull off and type in **“Execute Trace”** and hook this up to your desired input key (**If using enhanced input,** I recommend putting this on started **NOT** triggered, however standard key inputs work as well as shown in the image below)

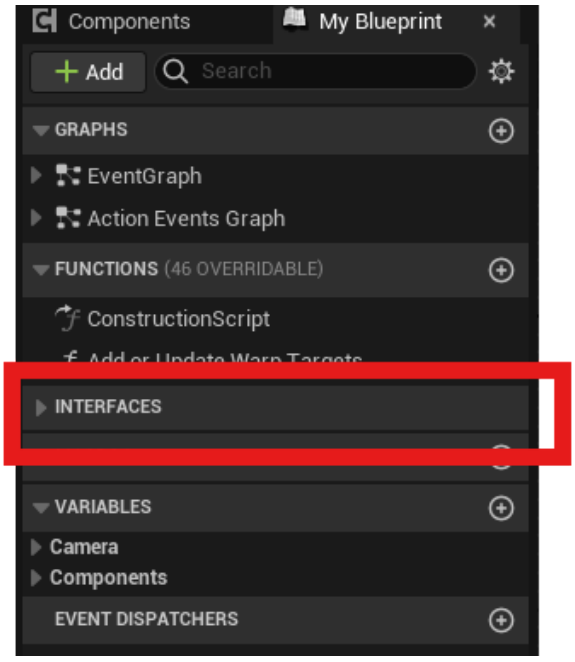




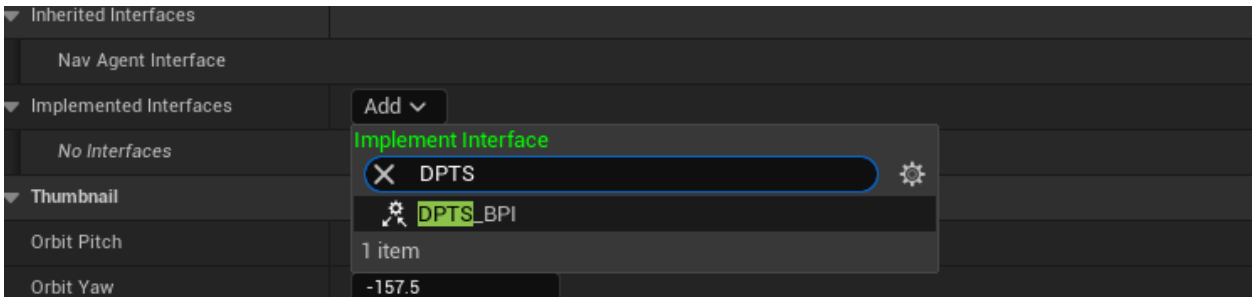
- While in your character blueprint, go to the top of the screen and select the box that says **“Class Settings”**



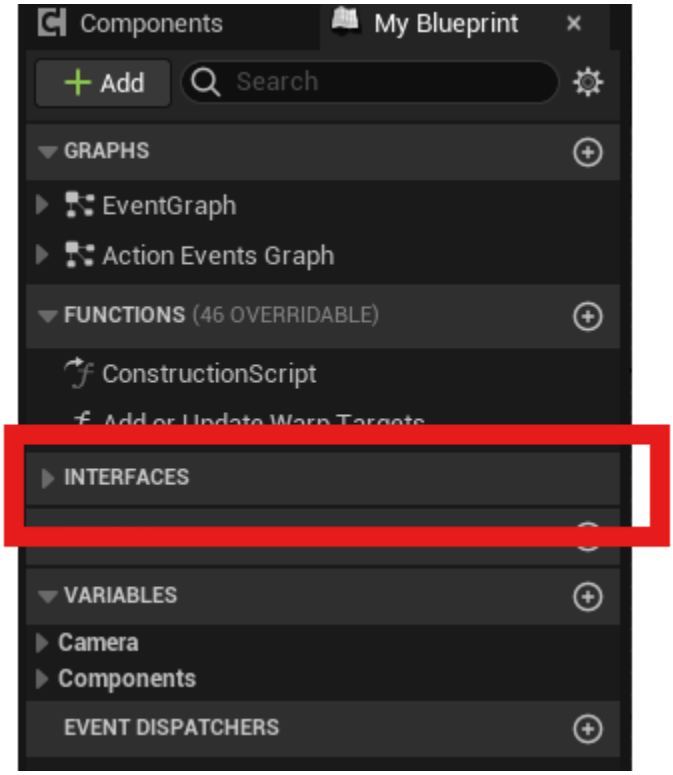
- In the Details Panel on the side, look for a drop down that says **“Interfaces”** and click on **“Add”** under **“Implemented Interfaces”**



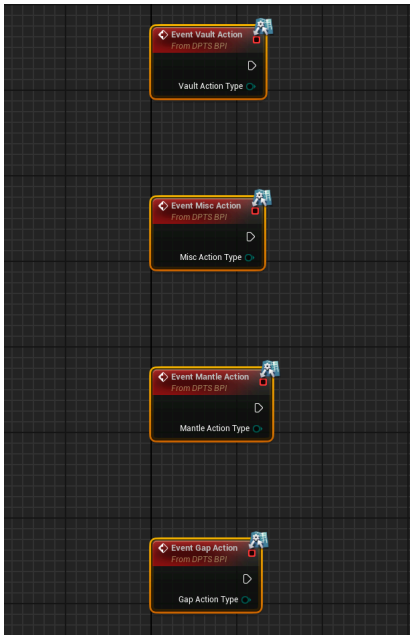
- Search for **DPTS_BPI** and select it to add to your character BP



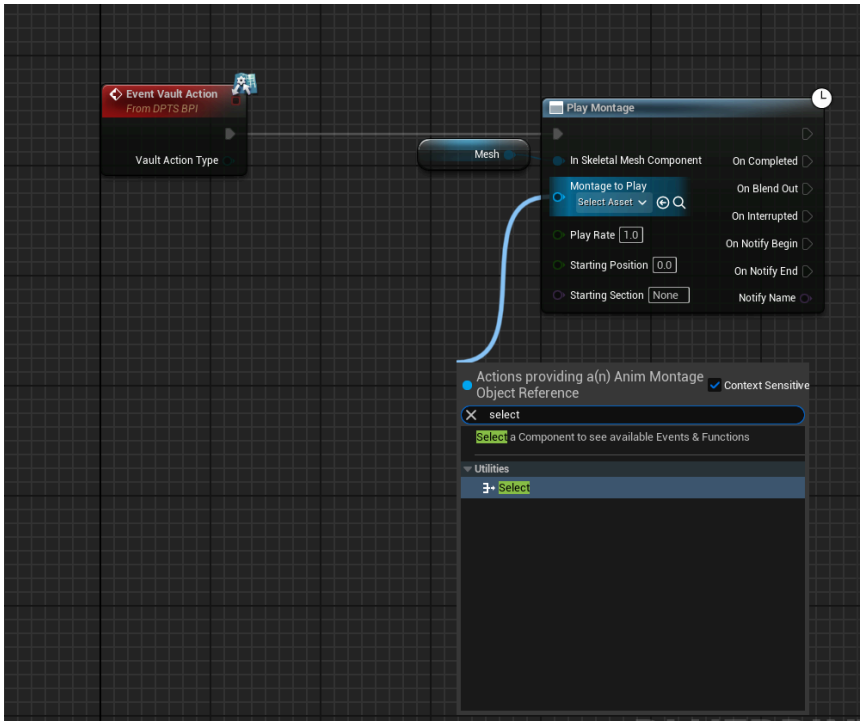
  - Once added you will have a newly created tab in the **“My Blueprint”** tab called **“Interfaces”**



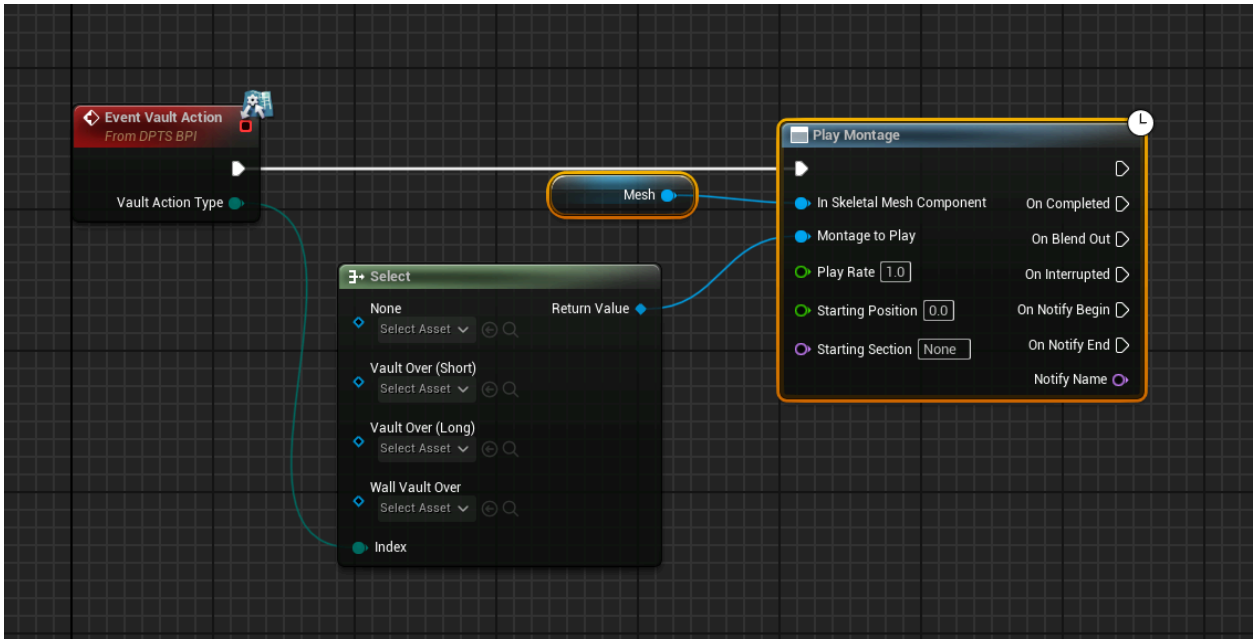
- Click the drop down to view the full list of **“Action Events”** and double click on the action types you wish to use to add the events to your graph



- Add a **“Play Montage”** node for as many executions as you have added, with the mesh ref hooked in as you would normally, and connect the execution pins. Then pull off the montage drop down on the node and type in **“Select”**

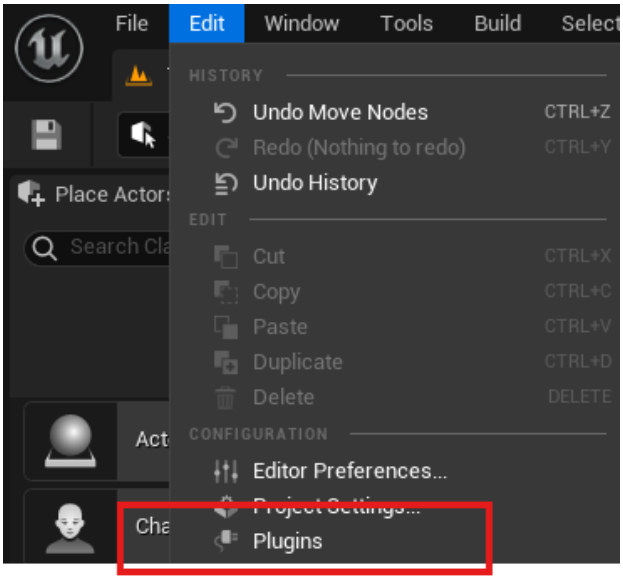


- Connect the enumeration output to the **“Index”** pin of the Select Node to give you montage slots for each action type that the action category has and add your montages to their matching action type (If un-familiar with how to make montages there are several videos and resource material on this subject online)

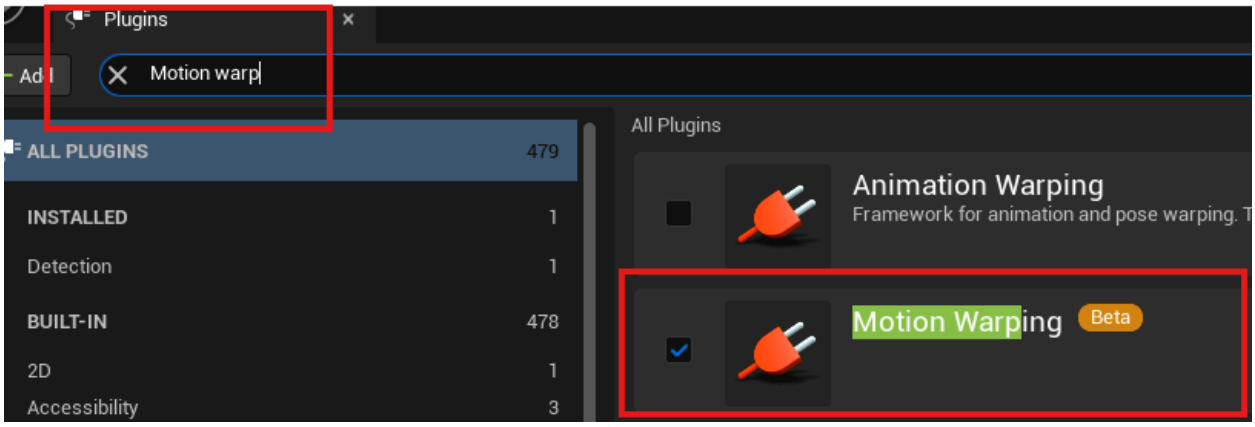


Optional Step if Using Motion Warping (Recommended)

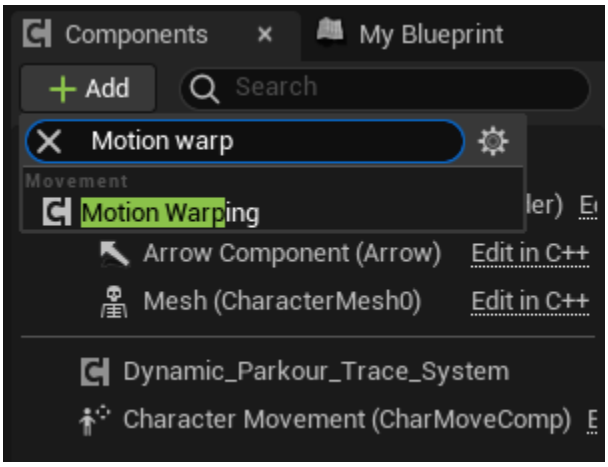
- Enable motion warping plugin by going to the top of your editor window, click **Edit** then Click on **Plugins**



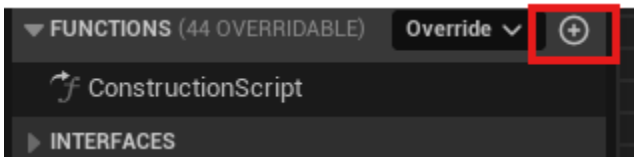
- Open up the plugin menu and search for and enable “Motion Warping” and restart your editor



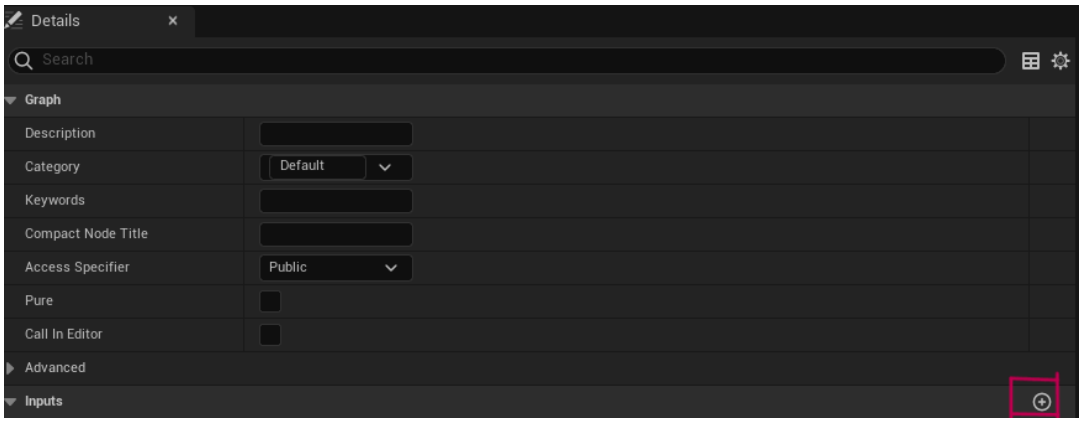
- Add the Motion Warping component just how you added the “Dynamic Parkour Trace System” component by clicking “Add” under components in your character blueprint

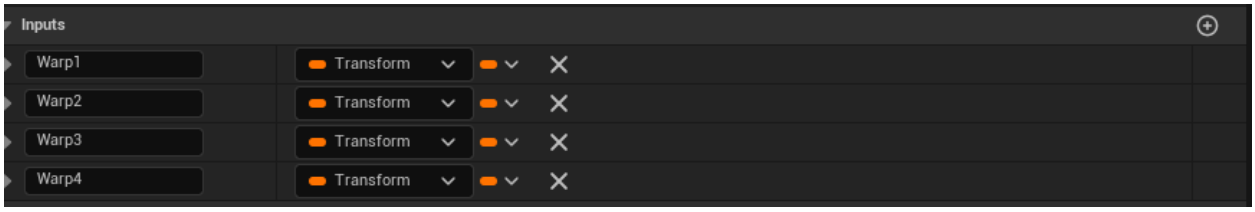


- Create a new Function by going to **“Functions”** in your character blueprint and select the **+** symbol and name it something like Add or Update Warp Targets and open it up

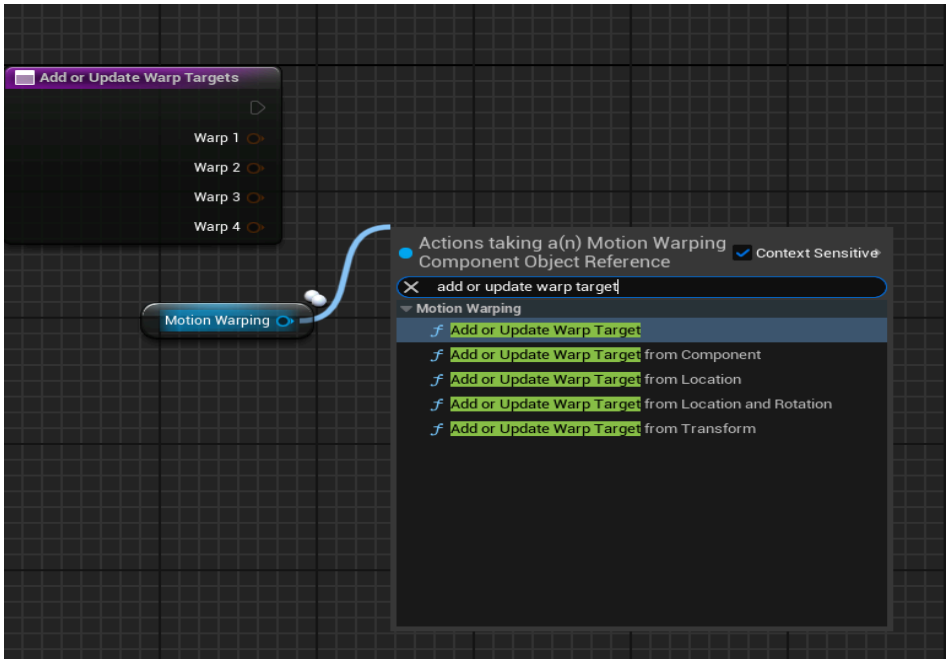


- Add 4 inputs to this function and set the variable type to Transform by going to the details panel and clicking the + symbol under **“Input”** and name them as shown below (This can be changed, however the system sends locations back as warp1 warp2 warp3 and warp4 and is easier to understand which location is what if you name them accordingly)

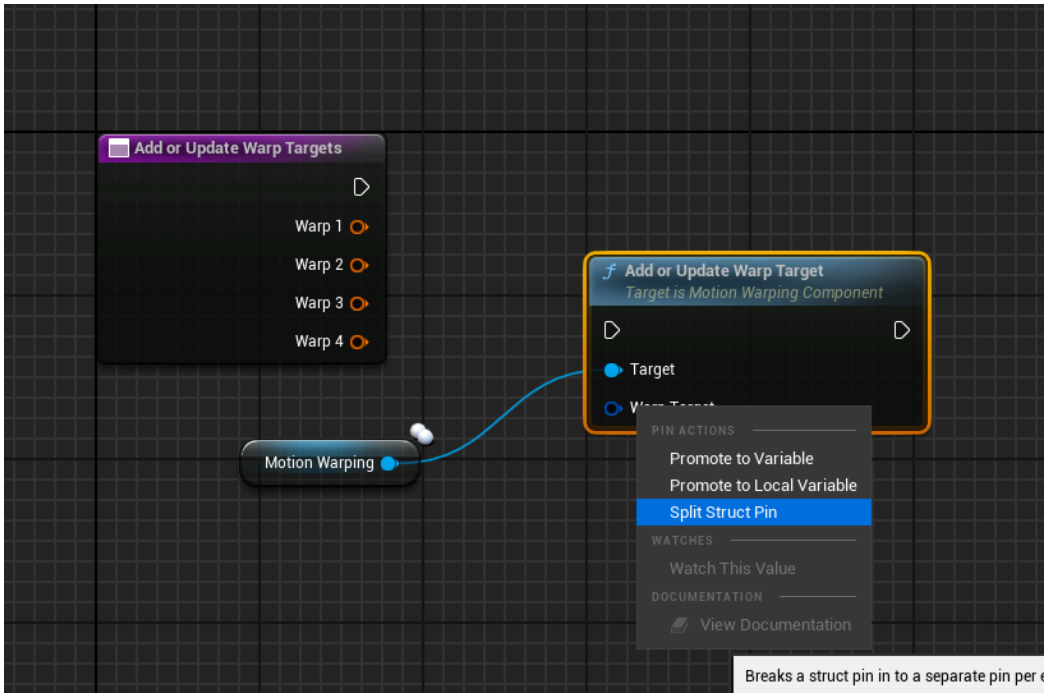




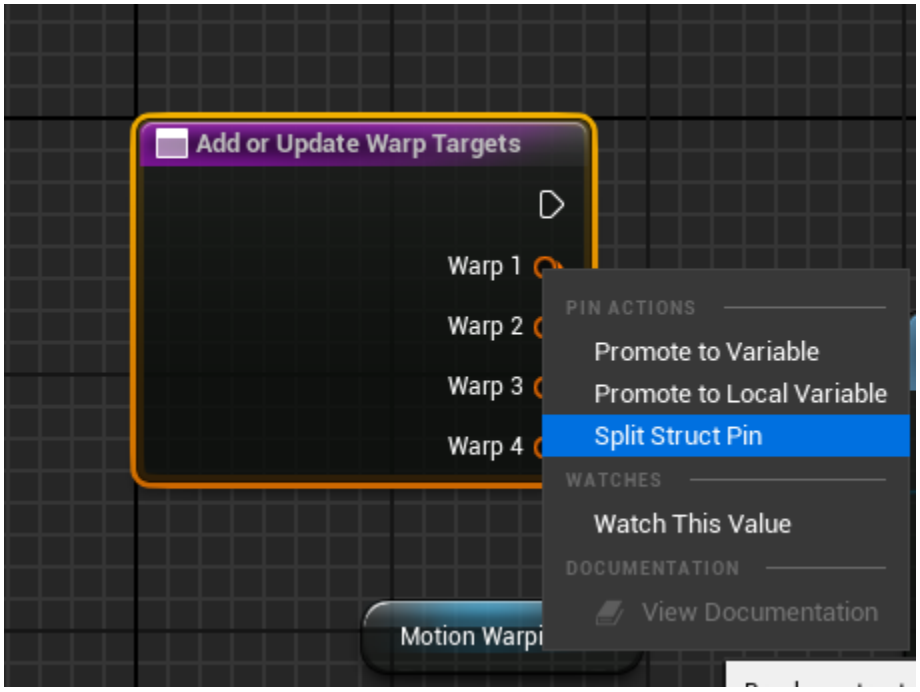
- Grab a reference to the motion warping component that you added earlier (Click and drag from component list) and drag into the function, pull off the reference and type in **“Add or Update Warp Targets”** and press enter



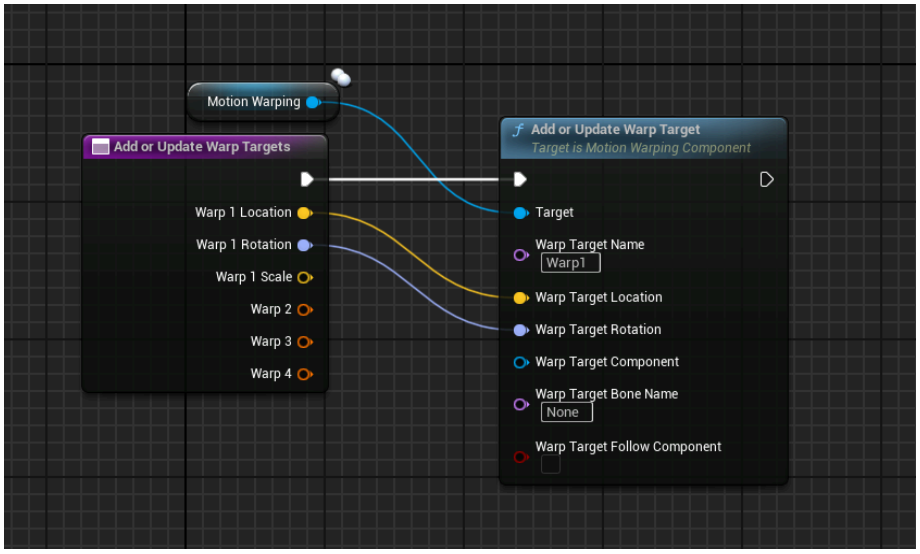
- Right Click on the **“Warp Target”** input in and select **“Split Struct Pin”** to expose the needed information



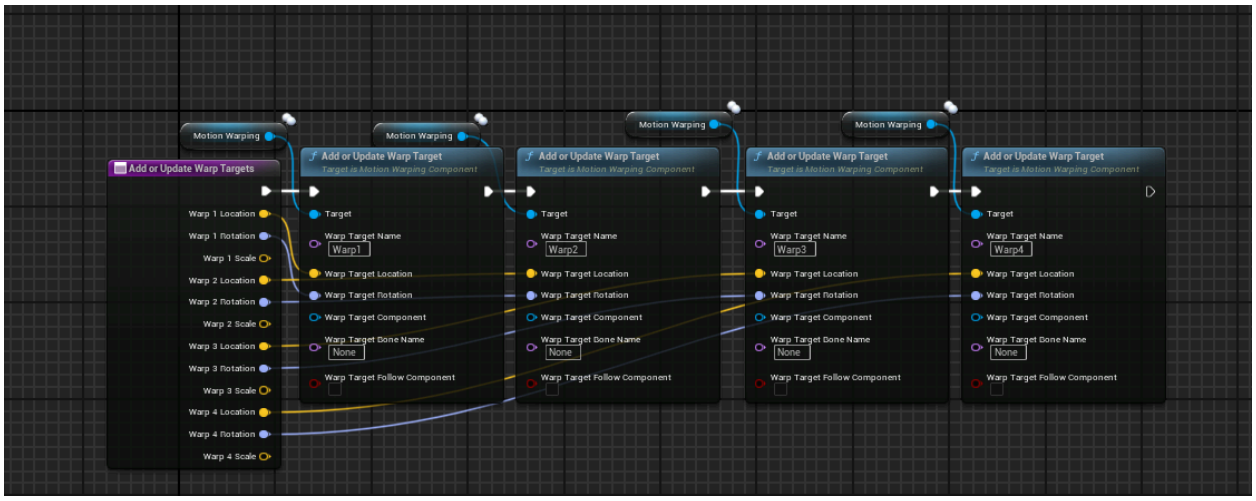
- Then do the same thing on the **“Warp 1”** transform input pin on your function and split this as well



- **The Update Warp Target** node will have several inputs, but only need Location, rotation, and warp name to be filled in. As stated before naming things consistently does help but you can name these whatever you want as there is nothing in the system controlling warp target names but it is recommended to set warp name to same name as transform pin being used, once **Warp name** is set, plug the split warp 1 l**ocation** and **rotation** into the add or update warp target node

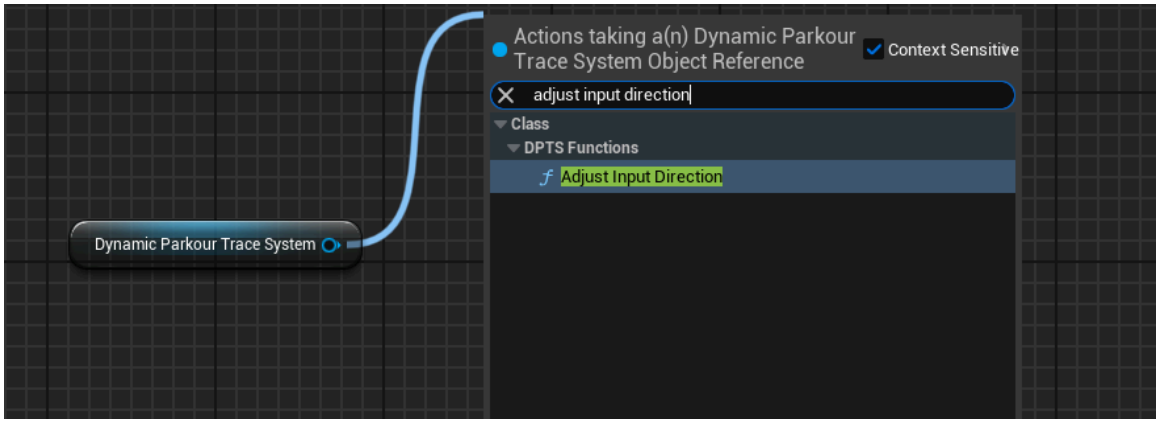


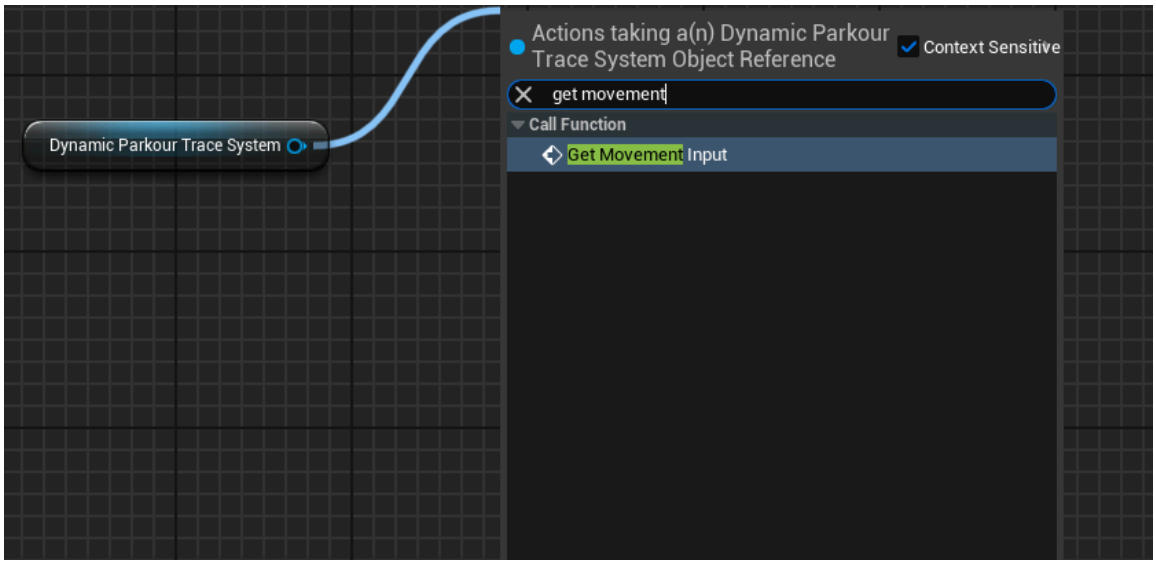
Follow this step 3 more times, naming them Warp2, warp3 and warp4 (unless doing self made names then its whatever you want) and then plug the location and rotation pins into their respective **“Add or Update Warp Target”** node and connect all execution pins together



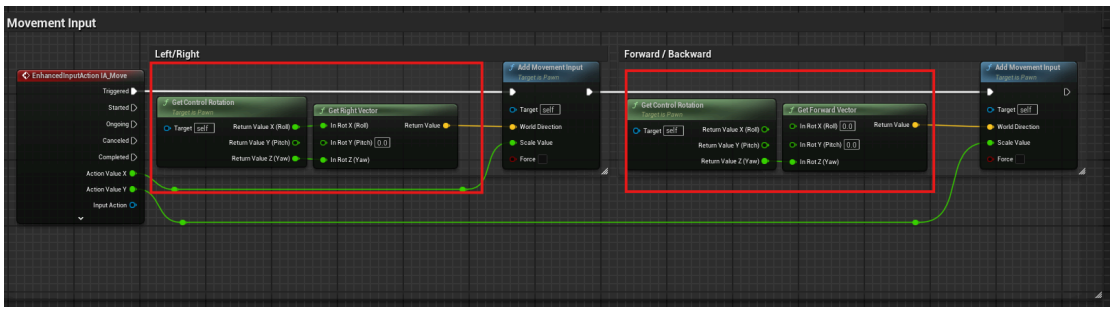
Additional Set up if Using Climb / Ledge Shimmy

- Get a reference to the **“Dynamic Parkour Trace System”** into your player character graph, pull off and type in **“Adjust Input Direction”** - Get another Ref (Or you can use the same one as these are going to be placed on the same execution line) pull off and type in **“Get Movement Input”**

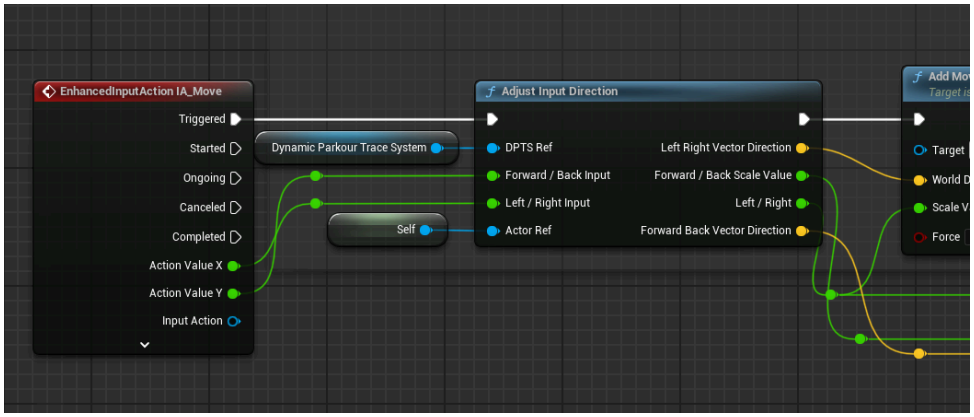


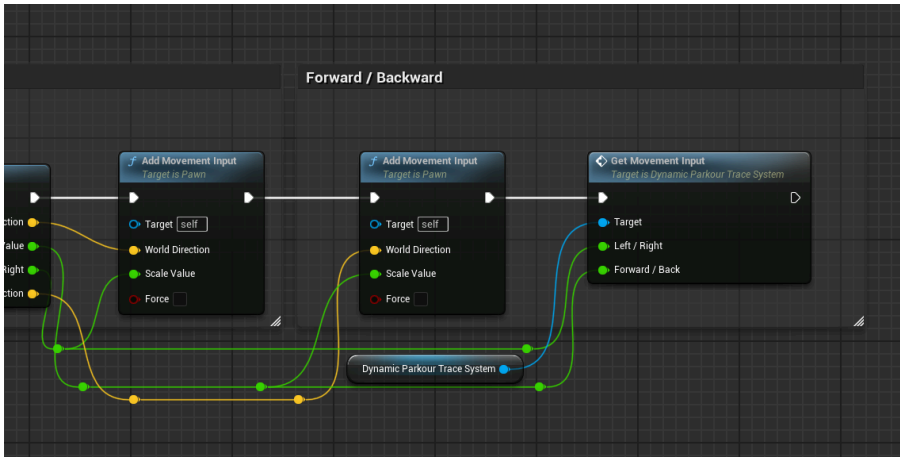


- Go to where your movement input is executing in the event graph (if using template character) and remove direction / rotation nodes highlighted in the image below

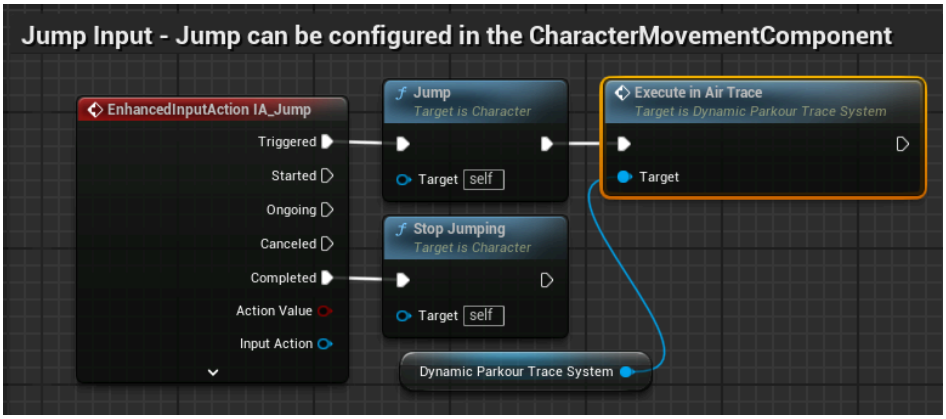


- Add the **“Adjust Input Direction”** function to the start of execution, and hook the X value into **“Forward Back”** and the Y value into **“Left Right”** and then pull off the pin that says **“Actor Ref”** and type in self to get a **reference to self**, then add the **“Get Movement Input”** function to the end of execution, once again hooking the value of X into Forward / Back and Y into Left / Right however ensure to use the output values provided by the **“Adjust Input Direction”** function

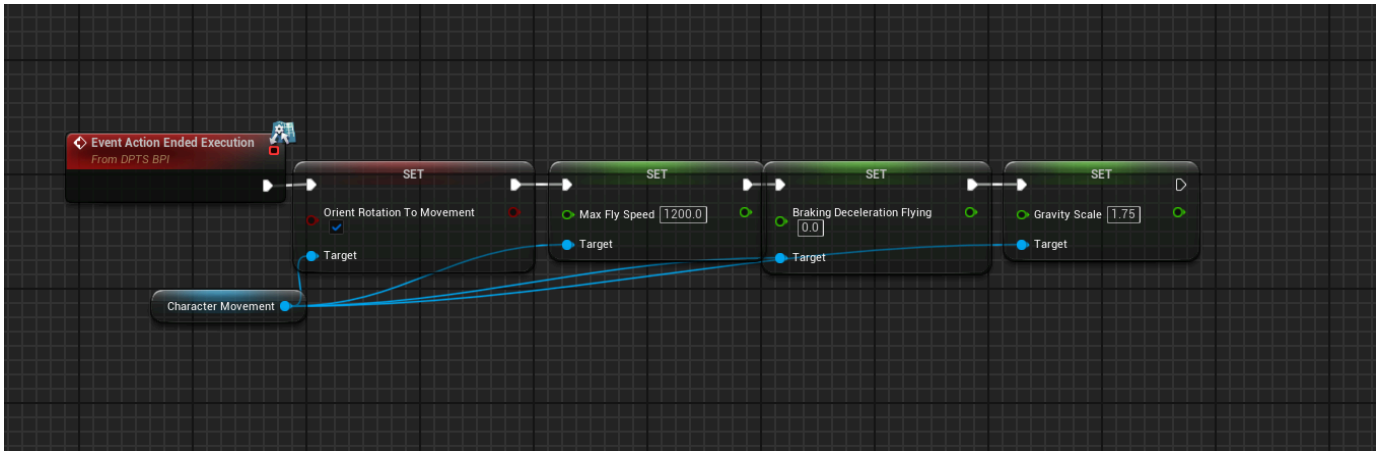




If using Wallrunning / Attach to ledge from Air Actions you will need to add an additional call after your primary “Jump” execution: Get a Reference to the **“Dynamic Parkour Trace System”** Pull off and type in “Execute In Air Trace” and attach this to the end of your jump execution to trigger the **wallrun and search for ledge** trace functions



System disables all rotation control values when in a climbing state, to ensure that all values can be restored to the developers needs upon action end, there is a new event that can be called in to reset rotation values, fly speed, gravity influence, and any thing else you may want to ensure is restored as this fires at the END of execution to ensure no overlaps. This event can be found under “Primary Events” in the functions tab and is called “Action Ended Execution”

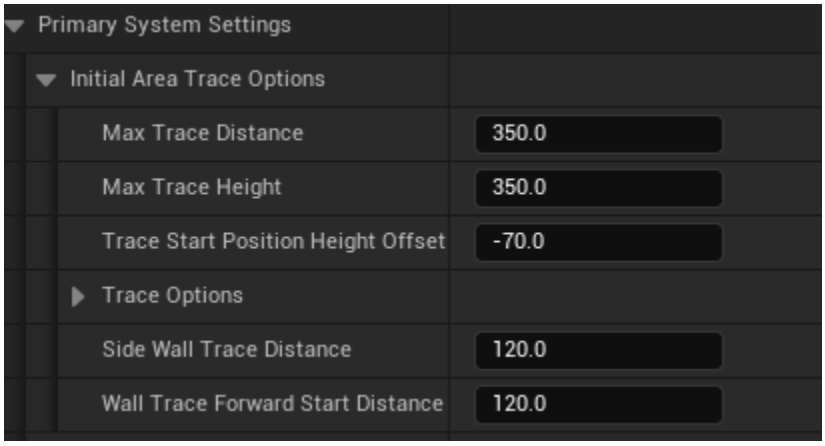


NOTE: Most values are reset upon action end such as plane constraints, movement mode, and collision settings - gravity influence is reset to engine default of 1.75 at the end of action but can be overridden using the above mentioned execution

Initial Settings Overview

The system uses a variety of traces to know what the world around the player is and what type of actions can be performed, however you can modify these allowed actions and trace behavior using the settings menu - Located in the Details Panel of the Dynamic Parkour Trace System component. To Modify the trace, go to the tab labeled **“Primary System Settings”** and click the drop down to expose the various trace / actions settings that control how the system defines certain action availability.

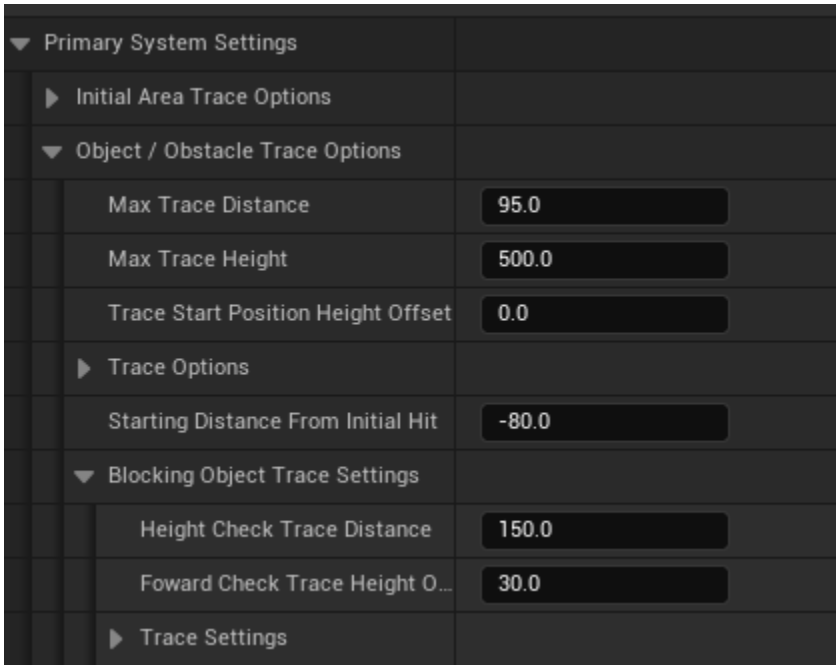
Initial Trace Settings



This menu can be used to define how high up and forward the initial trace will run to search for traversable objects, as well as settings a starting offset from actor position to ensure lower object detection - System also looks for possible side walls and the 2 bottom values will control how the system traces for said side walls (Tictac action primary use)

Most menus with unique actions / Trace information will also have a **“Trace Settings”** tab allowing you to define different looks or draw times for traces when debugging

Not all trace information can be found in these two settings menus however. The system consists of 3 main parts - First traces the world looking for ledges, edges and objects. Second traces that found objects height and depth to get object dimensions, and then all data is run through a series of calculations to defined what the most appropriate action should be. The values that control this can be found under **“Object / Obstacle Trace Options”**



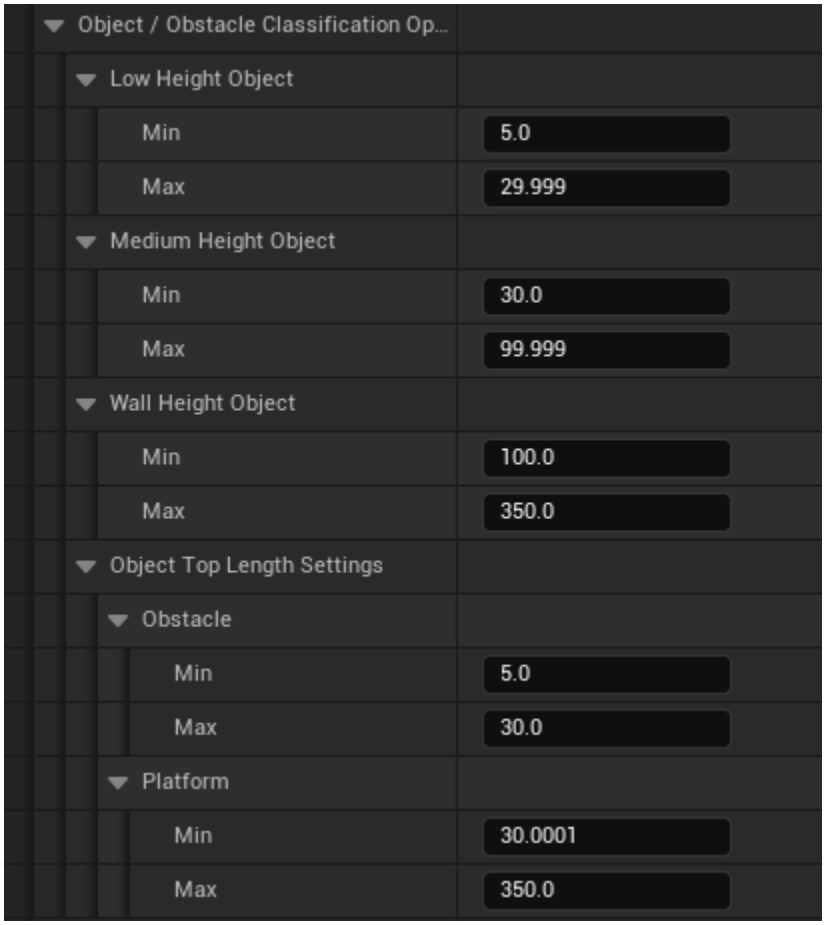
These values control how the system traces the found object but uses the found initial vector to start its trace position - Use the starting distance from trace offset to pull the start of the trace back if using complex collision items.

Max trace height is used to define how tall an object can be before it's considered out of reach.

System also has traces that go up from found height position as well as a forward trace that runs from height point to depth point to ensure that there are no **blocking objects** in the way of performing an action - If you have world filler items in your game, **make sure to disable the collision channel response to whatever you are using for traversal to ensure that small objects do not block action availability.**

Max trace distance is how far forward trace runs starting from a combination of the initial hit vector, and settings starting distance value. If you pull the hit back or forward it is recommended to also adjust the distance for the trace

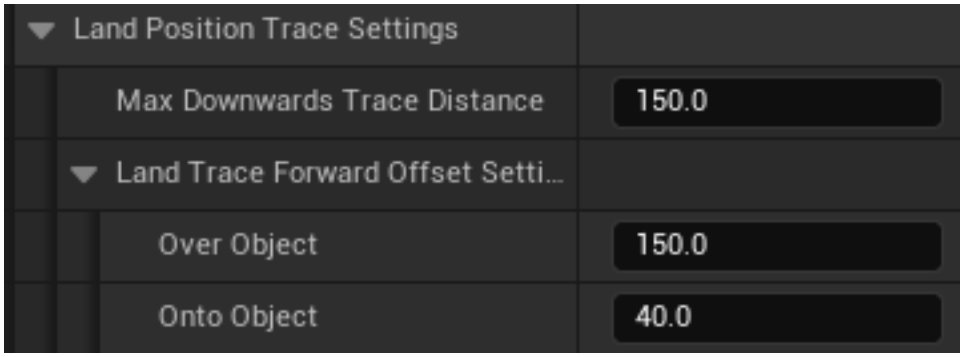
After tracing the object the system uses the values defined in the **”Object / Obstacle Classification Options”** menu to choose the most logical action based on the defined parameters. Use these values to fine tune how actions will be chosen (Hover over the Settings you are adjusting to see what actions this value influences)



Obstacle / Platform is used to define **“onto / Over”** type action choice

Ensure that when adjusting these values you put the value with .999 or .001 to ensure no gaps in distance detection

Action landing settings can also be adjusted in this menu by adjusting the values in the **“Land Position Trace Settings”**

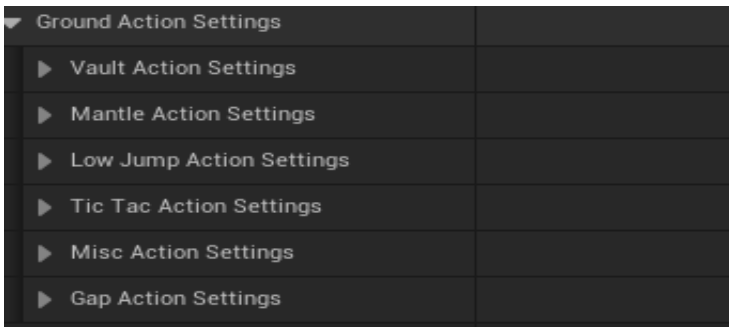


Downward trace amount will provide back land location based off hit or trace end to ensure that when no ground is found a location is still provided back when vaulting over higher objects into a falling state

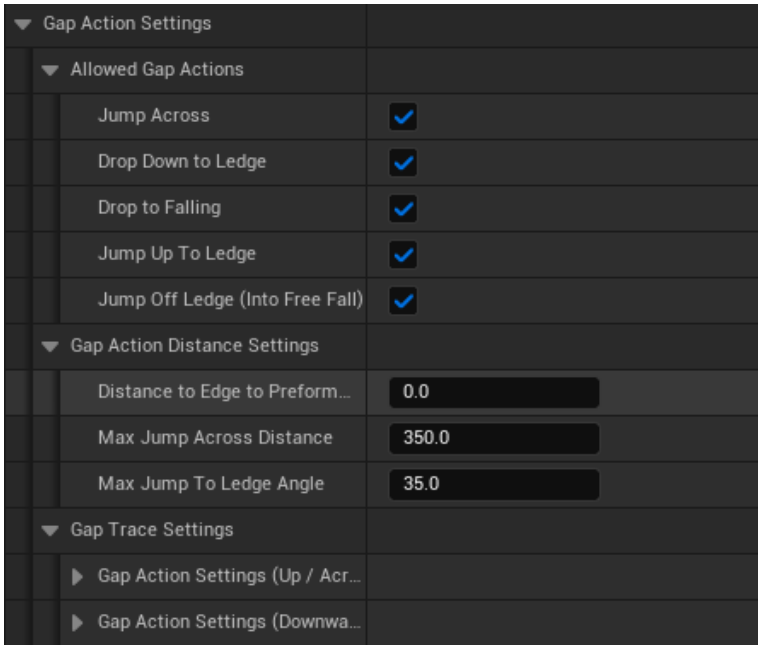
Land Trace Forward offset settings control how far forward the trace will start from the depth position to find a landing vector - Use onto and over values to control forward landing position for onto / over action types.

Not all trace settings / action definition settings are controlled through these menus - Certain actions have specific overrides and these are built into the corresponding action type menu / action type setting (E.G climb actions in climb settings, ground actions like vault, gap, mantle, tictac ect are found in ground action settings)

To Locate the additional action / trace settings for non-climb / ledge shimmy type actions, click the drop down under **“Ground Action Settings”** and you will have access to a variety of settings controlling different actions. Most actions are a simple boolean list of allowed actions as other settings already control how they operate (**Mantle** actions for example are controlled using the settings defining **platform / obstacle** and object height settings of **medium or high**)



However there are certain actions that have their own values controlling their specific action calculation like gap actions for example.

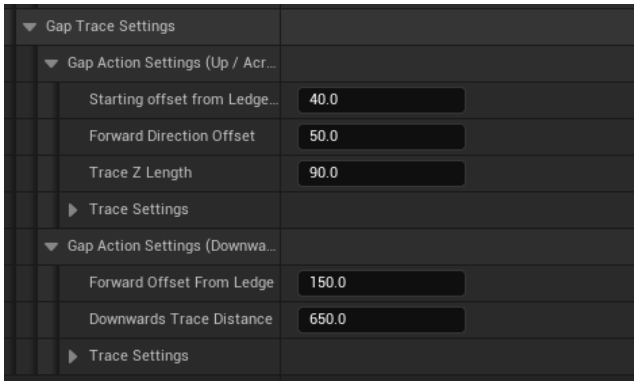


The Allowed Actions list will block any action from executing that is not checked

Settings such as **“Max Jump Across Distance”** control not only the actions availability but also the distance as which the trace runs to find a possible jump across / up to object.

Max Jump to angle is a limiter based off the necessary Angle of Attack to complete the action.Put simple the farther away it is, or the higher it is in relation to the player, the higher the value needed to complete a jump to ledge type action.

Gap Trace has two points - Up / Across and Downwards - Use the values under “Gap Trace Settings” to adjust these to your needs

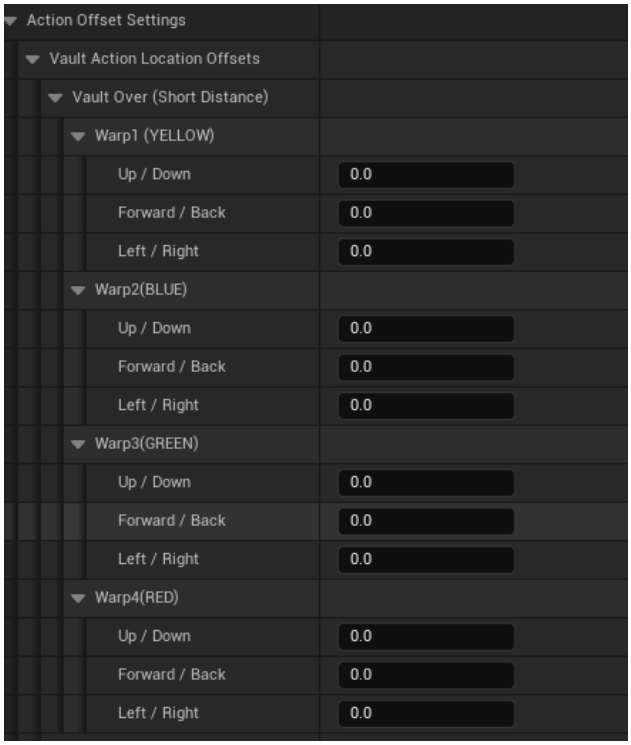


Tic Tac Actions Also have a controlling value in their menu under “Ground Action Settings”

TicTac Actions are defined by this area of available height - To allow for lower or higher tictac actions, adjust these values as desired.

Please ensure that you are selecting ALL actions you wish to have be preformed by going and selecting them in their respective menus - If you do not allow an action, it will NOT perform it. Regardless of what other settings you have applied to the system.

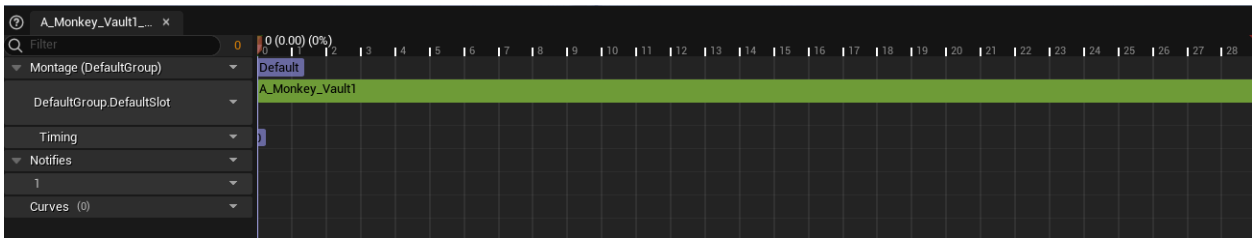
All actions locations can be adjusted using the “Action Offset Settings” Defaulted Locations are for UE5 manny scale type character



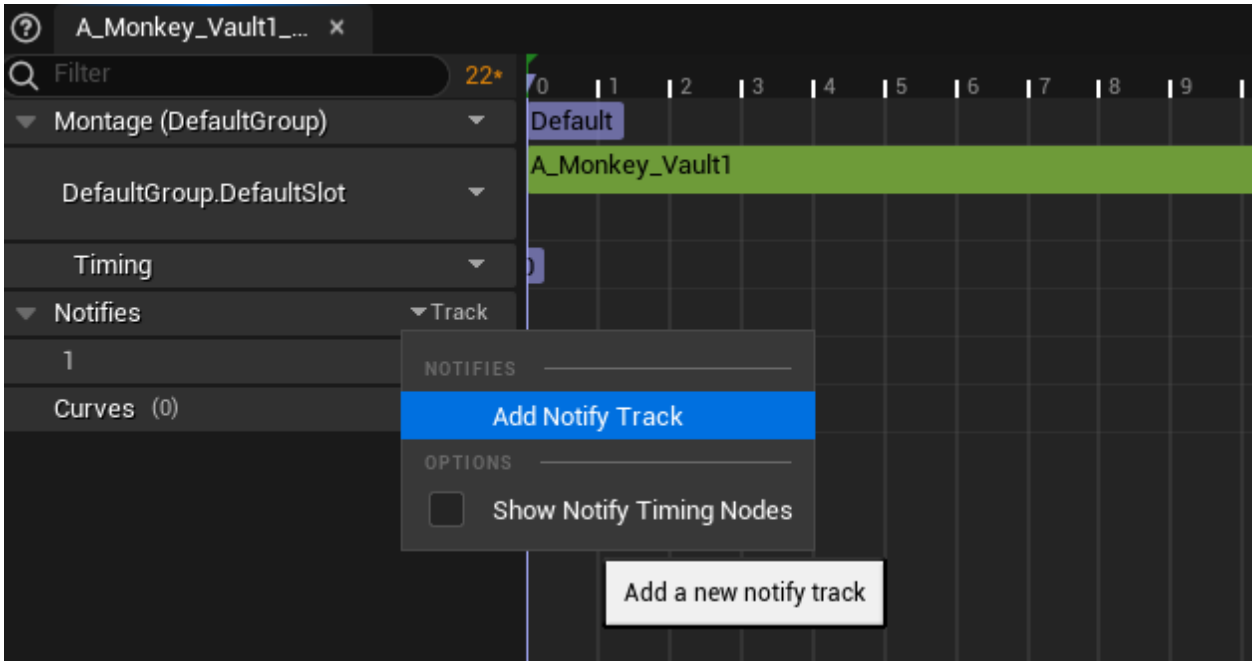
MotionWarp Setup

Once you have your montages created and root motion is enabled (Right click on an anim sequence in your content browser, and go to Create > Create montage) you will need to add Motion Warping windows to ensure the actions location gets corrected.

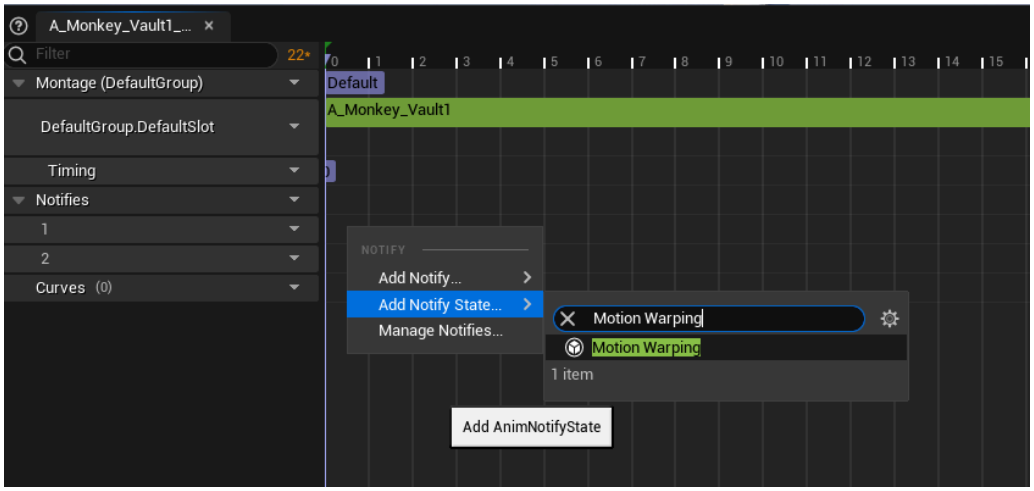
First open up any montage you want to use and you will see a series of tracks at the bottom



Go to where the notify track section is, click on the drop down arrow on the right hand side, and select **“Add NotifyTrack”**



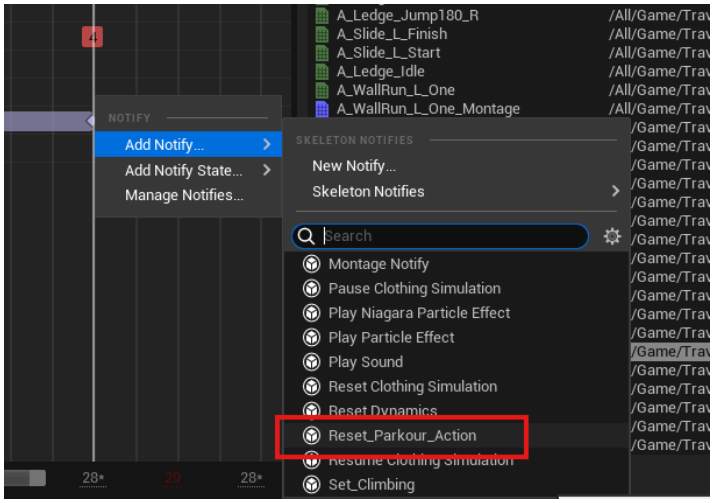
With 2 tracks available right click on either notify track go to **“Add Notify State”** and type in **“Motion Warping”** to add a motion warping window to your montage



This will move the actor during this window from where they are currently (First point) to the defined location matching the warp name defined in Add or Update Warp Target function made earlier (Last point) and you can use as many windows as you have locations (System max is 4 locations per action which is far more than enough as most you will really only need 2, or 3 at the most)

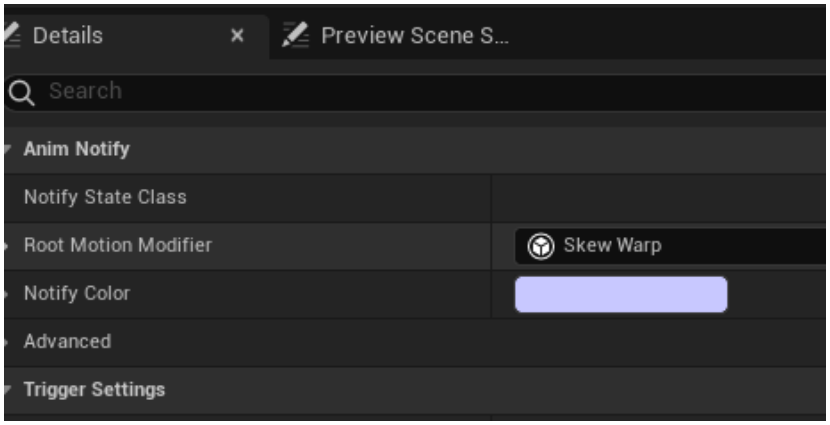
Motion Warping overrides root motion of the animation - As such you want to try to match these windows during the motion in the animation to ensure that the movements match the locations - for more info about refining this and how to determine the best window locations check out the tutorial on motion warping available in the playlist linked at the bottom of this document OR on the Unreal Engine Marketplace page for this asset.

Once windows have been created you need only add a pre-built notify to reset the action back to default - Right click again on a notify track, and go to **“Add Notify”** and select the option that says **“Reset Parkour Action”** and line up its trigger point with the final motion warp window

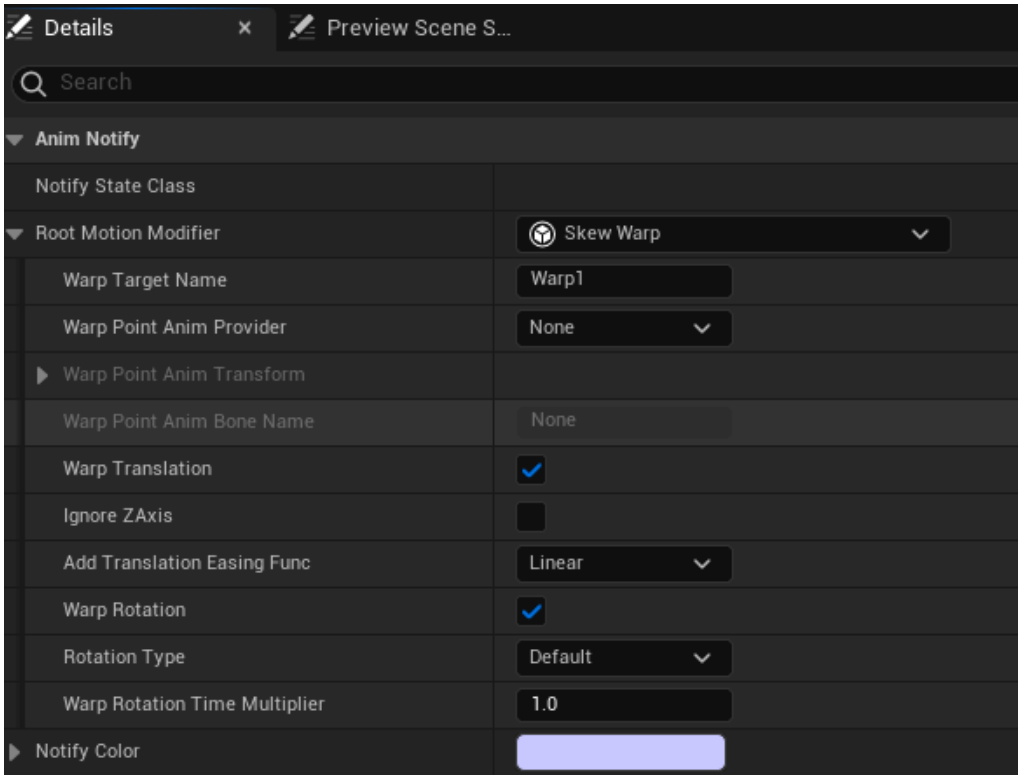


Lastly just need to set up a few settings in the **motion warp windows** that you created.

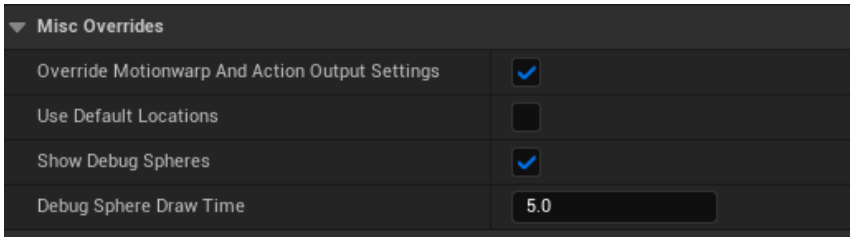
Click on the **motion warp window** to highlight it and go to the details panel on the right side (Default UI Set up) and click the drop down arrow next to **“Root Motion Modifier”** This will expand the motion warp window settings



Warp Target Name needs to be what you named the Warp1 input in the function **“Add or Update Warp Targets”** Made earlier - These have to be EXACT both in spelling and capitalization. It will then get the associated location and use that as the motion warp destination. Also make sure to Uncheck **“Ignore Z axis”** otherwise there will be no upward movement to correct height when needed



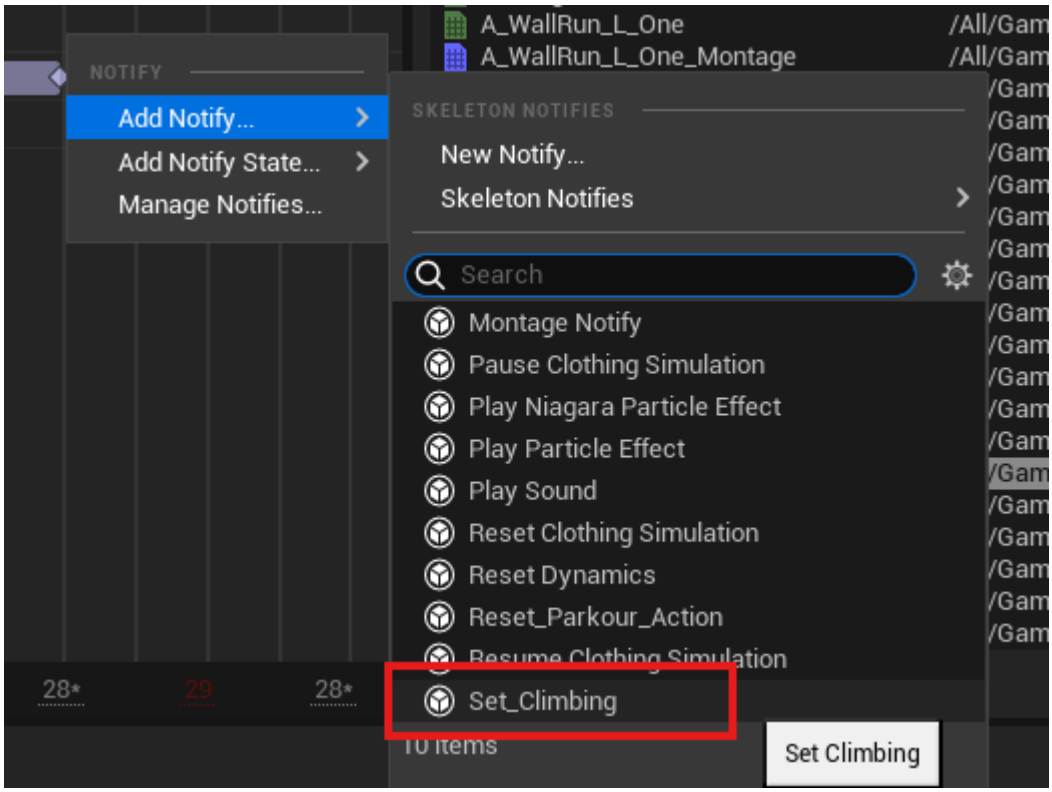
Do this step for each motion warp window created, matching the names to the desired location to be used for its target destination. (**Warp1 is Height, Warp2 is Depth, Warp3 is Land, Warp4 is misc** used differently for different actions) To assess what locations are best to use for the animation you are using, it is recommended to open up the misc settings menu, and select **“Show Debug Spheres”** and **“Override MotionWarp / Action Output Settings”** - Debug spheres are colorized with matching color names to make it easy to assess which location belongs to which sphere.



<u>Not All Actions Use “Warp Rotation”</u>

If the Action ends on a ledge / Climb Style Action make sure to uncheck "Warp Rotation" Excluding Wall Climb to Ledge!

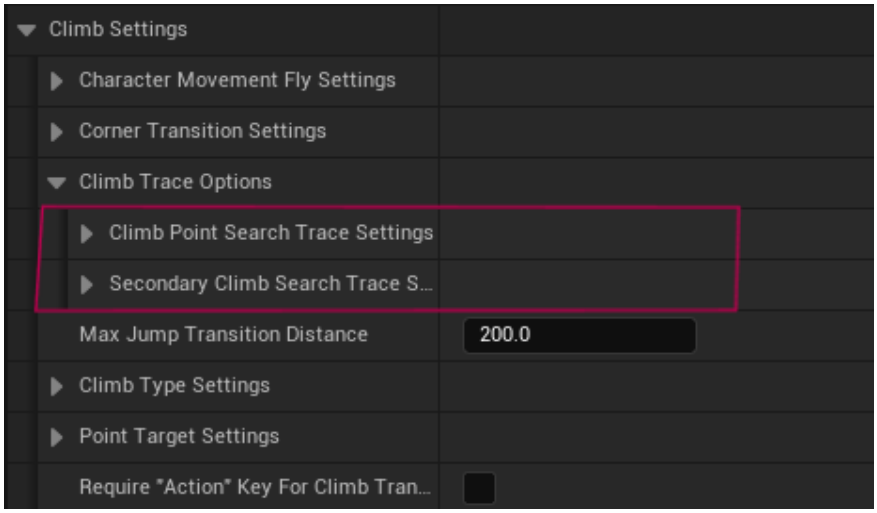
IMPORTANT!!: If Using action that initiates a climb (Lege, Free Climb, or Point to Point - Does NOT matter) the end notify added to the montage changes. Follow the same steps as above except replace "Reset Parkour Action" Notify with "Set Climbing" to ensure that proper settings are adjusted for climb type actions - Any action that ends climbing (Drop down, climb up ect) follows the steps mentioned above to reset



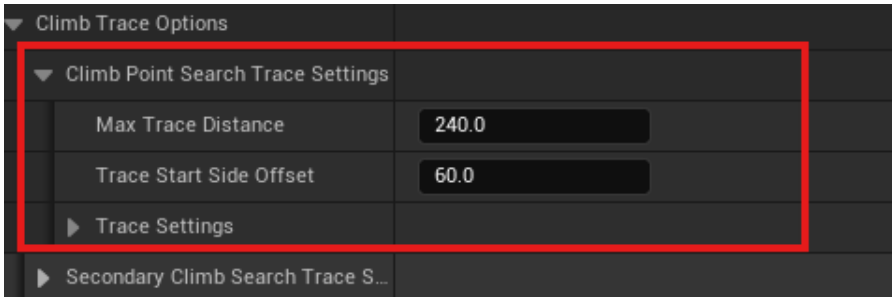
Climbing Settings:

Trace / Allowed Actions Settings

Climbing is comprised of two main elements - First is tracing for available ledges found on world objects - Second is using **“Climb Point Generator”** which can be found in the BP section of the **DPTS_Assets folder** (See Climb Point Generator Settings and Details for more information)
System uses a series of two different traces to confirm which type of ledge / climb point it has found and the settings for these traces can be found under **“Climb Settings”** and its drop down menu **“Climb Trace Settings”**



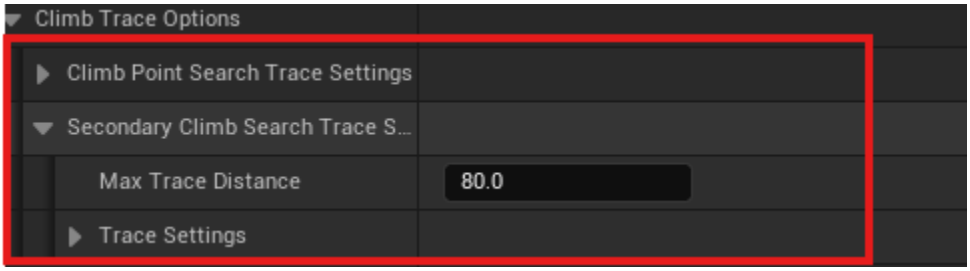
Climb Point Search Trace Settings Controls the systems trace for climb points while in a climbing state



Max Trace Distance controls how far the trace will go to find a possible climb point generated by the climb point generator
T**he Trace Start Side offset** controls how far left or right the trace starts to help avoid overlapping on current climb point

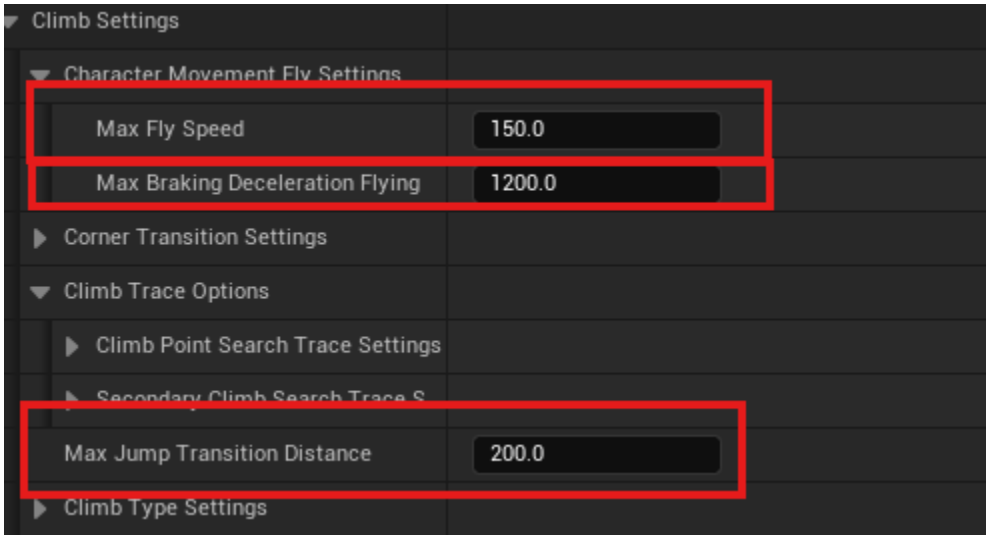
Secondary Climb Search Settings:

The Secondary Climb Search Trace Settings control certain trace values when looking for a ledge nearby the actors position that is not a generated item to jump to or check if the current object can be climbed onto



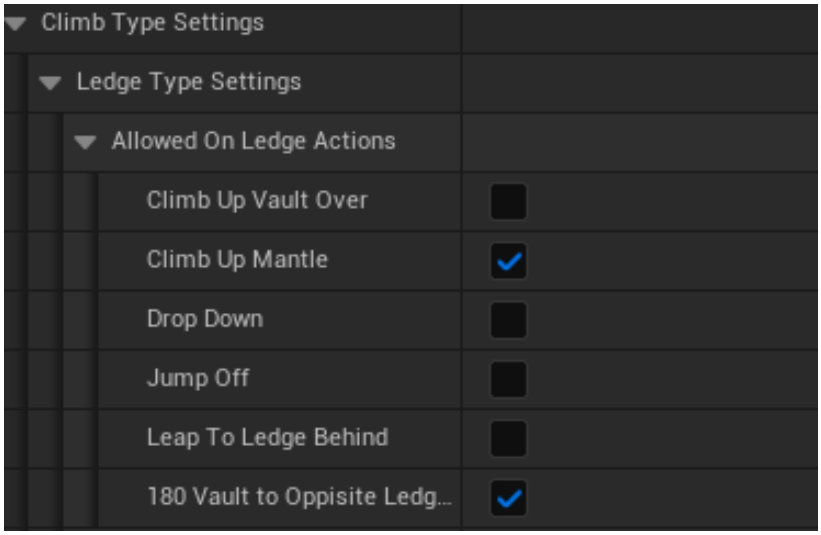
Max Trace distance controls how far forward the trace will run from the current ledge height when searching for a non-climb point ledge

You can also control how fast the on ledge / free climb movement happens and how far the actor can climb jump to by going to the **“Character Fly Settings”** and adjusting the two values of **max fly speed** and **braking deceleration flying** (Higher speed = faster ledge movement, High deceleration = Faster stops on input stop and vise versa on opposite value adjustments) and the value “**Max Jump Transition Distance”** highlighted in the image below

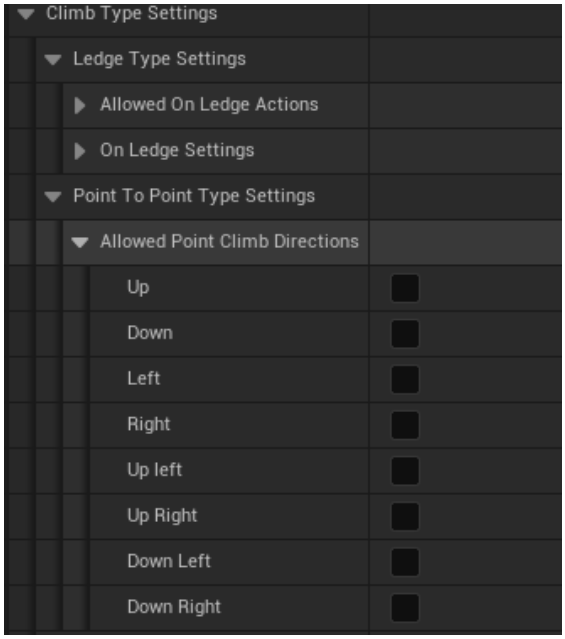


Ledge / Point Climb Settings:

Climbing Also has various actions that can be performed during certain climbing states such as **90 corner transitions** (Inner and Outer) as well as **climb up, drop down, jump off**, and various others. To enable or disable these there are allowed actions menus that can be found Under **“Climb Type Settings”** and the respective action drop down will contain settings for this action as well as an allowed actions list

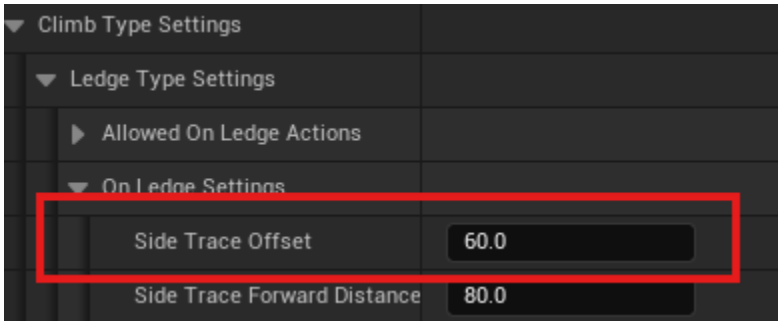


By default **Climb Jump** is 8 way directional - If desired certain directions can be disabled (Also helps with side scroller / top down games use) - This option can be found under **“Point to Point Type Settings”** then **“Allowed Point Climb Directions”**

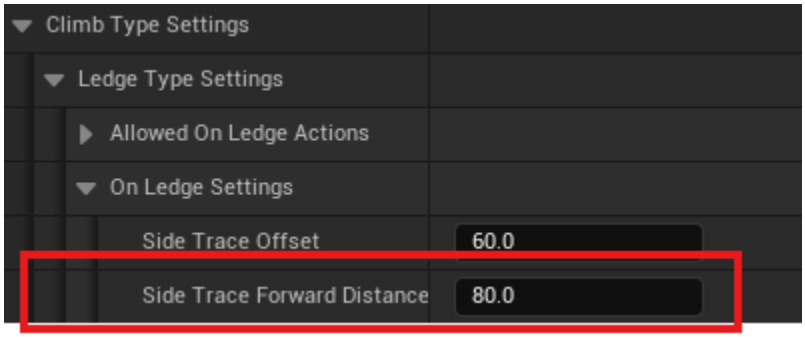


On Ledge Trace Adjustments:

To adjust how close the actor can get to a ledge corner, you can adjust the value **“Side Trace Offset”** Under **“On Ledge Settings”**



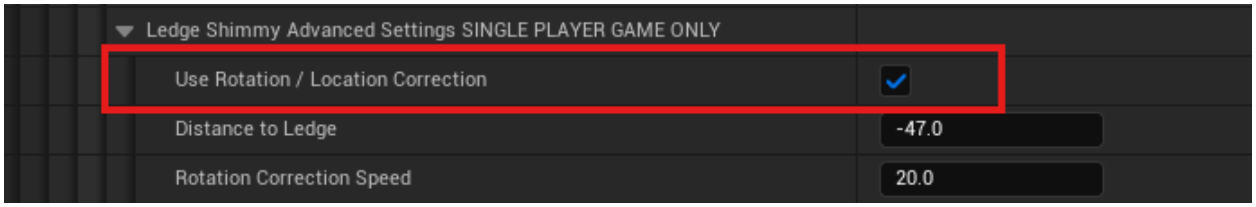
The Side Trace Forward Distance is how far the trace will run to look if the actor is still on a ledge (Also influences corner transition starting point to ensure that ledge corner transitions are influenced smoothly)



Ledge Shimmy Advanced Options:

SINGLE PLAYER ONLY (for now)

For single player games there is an additional option to provide rotation / location correction to the actor to ensure that they are always facing the correct direction even on curved surfaces - To enable this setting click the drop down menu under **“Ledge Shimmy Advanced Options”** and check the box named **“Use Rotation / Location Correction”**



To set how far the actor will be placed from the ledge - Use the **“Distance to Ledge”** value (based off forward vector math: So forward adjustments are positive additions, backwards adjustments are negative additions)

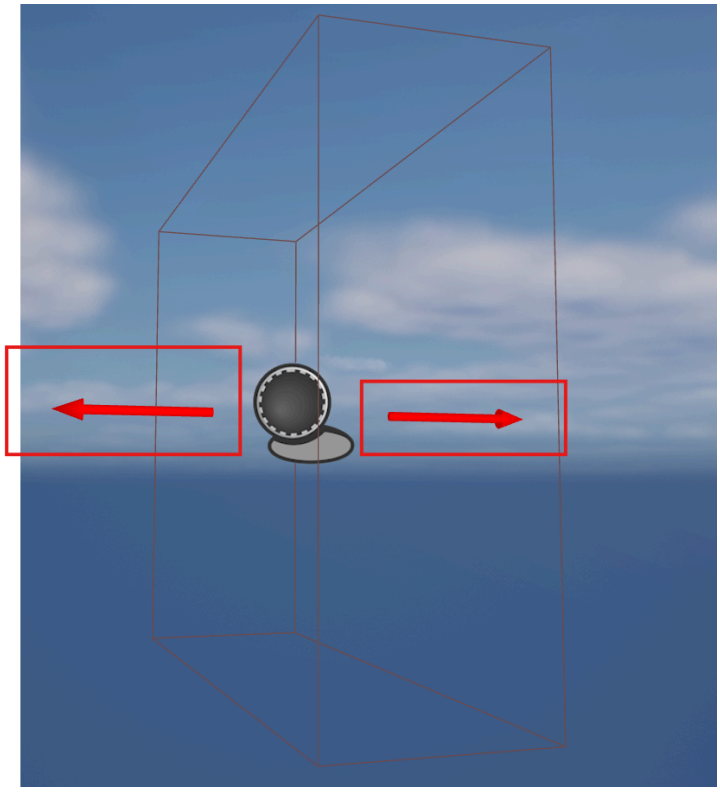
And you can also adjust how fast the rotation correction happens (Using **Rnterp** node and value represents the interp speed value) using the **“Rotation Correction Speed”** setting - Higher values will result in faster rotation

Using Action Key for Climb:

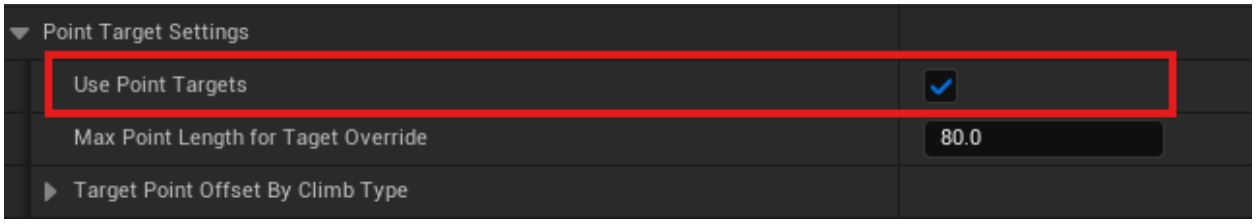
Climb movement can be handled one of two ways - First is free flow and will jump to, down, or preform climb up, and corner transitions based off input direction and will not require and additional input to proceed with the climb action - If you wish to use the input that the primary system uses (This can be jump or any additional key - However the actions will be called from the Execute Trace call) make sure the box is checked thats named **“Require Action Key For Climb Transition”** and system will not allow a climb action to be completed until the input has been entered

Point Target Settings:

When using climb points generated by the **“Climb Point Generator”** blueprint (See Climb Point Generator Section for more details) there are two ways to get a usable location - First is using the trace system and the location it finds directly in front of the actor - Second is using spawned **“Target Points”** that are represented by arrows on the climb points themselves and using these as the desired starting location.

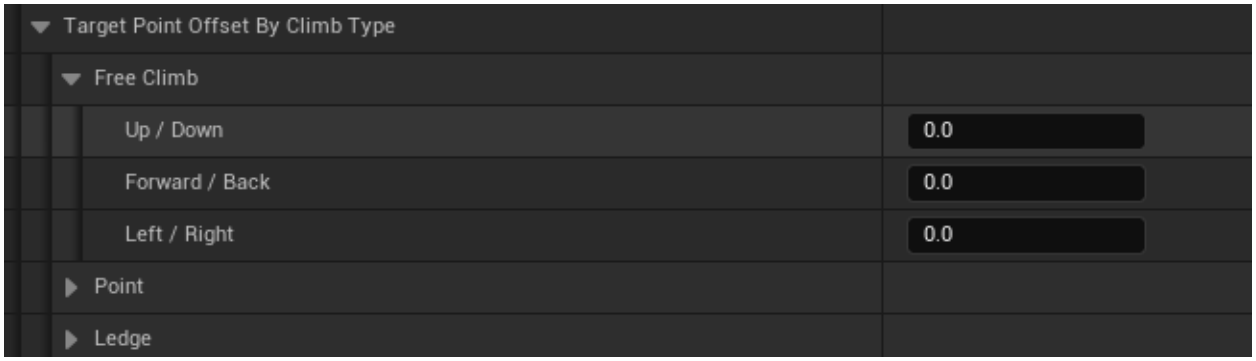


To enable this option - Go to the **“Climb Settings”** then expand the drop down menu named **“Point Target Settings”** and check the box named **“Use Point Targets”**



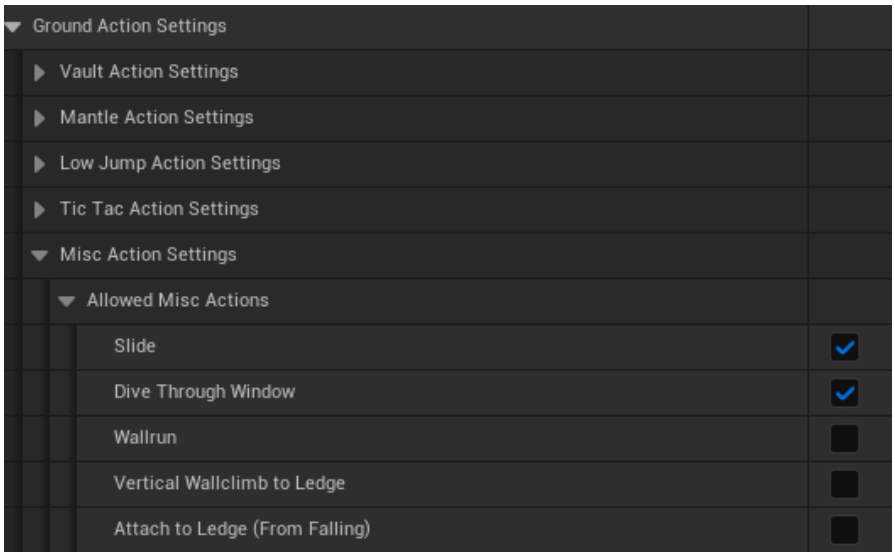
There is also an option to override the use of the target point if the found climb actors length is beyond a certain value - Represented by “Max Point Length For Target Override” (This is the **“Box Extent”** Y Value of the found point)

And if desired you can offset the point targets location using the menu “**Target Point Offset By Climb Type”** and each climb type has its own offset that will be added if using target points to assist with offsetting on more complex geometry (Applies to output location, NOT trace information)

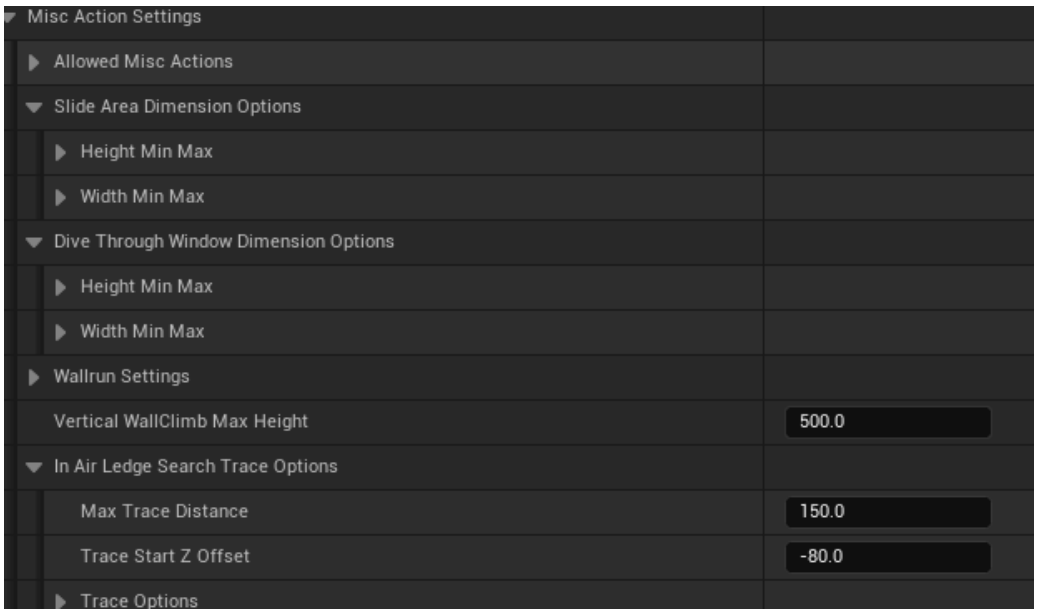


MISC Actions:

There are several actions that are provided into the system for some additional, unique movements, these consist of **Slide, Dive Through Window, Wallrun, Vertical Wallclimb to Ledge, and Attach to Ledge from Air** (Free Jump style ledge attachment) and these can be enabled or disabled using the **“Misc Actions Settings”** under **“Ground Action Setings”**



There are various settings for these actions as well to assist in how the system will allow / choose one of these actions



Slide / Dive through window:

These actions are defined by height and width allowances to adjust these settings for slide expand the menu named **“Slide Area Dimension Options”** and you can adjust the min / max height and width values for the system to allow a slide through type action

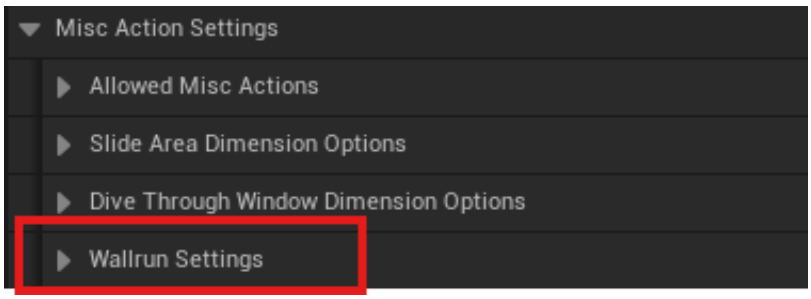
Same can be set for the dive through window / opening action type as well under its respective settings menu

There are additional actions that can be performed when a slide type action is detected, See **“Slide Modifications”** Under **“Modification Settings”** section of this documentation

Wallrunning:

To enable wallrunning you must first place a climb point generator with wallrunning selected as its climb type in the settings menu (See **“Climb Point”** section of documentation for more details) - This helps to ensure that wall running on incorrect surfaces doesnt happen - However you can place a wallrun point wherever you wish.

Once the wallrun area is created, you can find the settings for wall running under the **Misc Actions** settings menu, and expand the menu named **“WallRun Settings”**

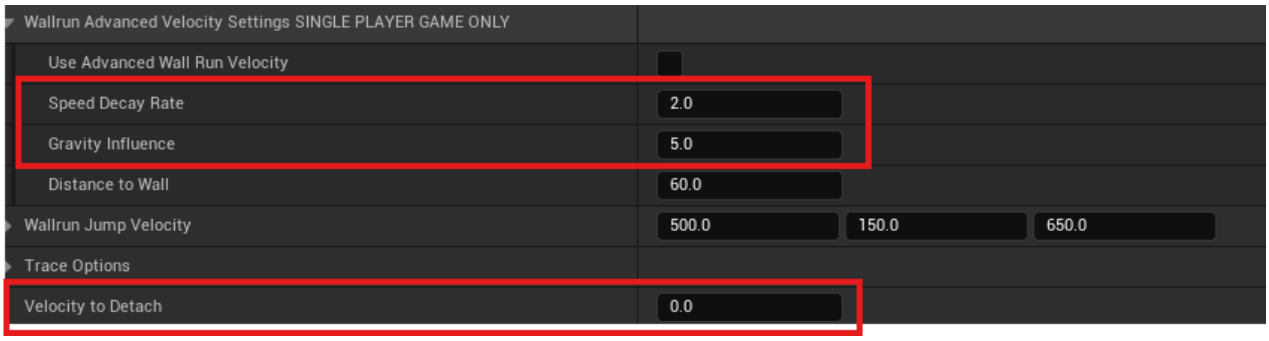


Wallrun action also has a wallrun jump action using a simple Launch Character Node, use the **“Wallrun Jump Velocity”** to set the launch velocity (**NOTE**: Set to 0 if you wish to use your own wall jump system) and use “Wallrun Speed” to set the speed at which your actor will start the wallrun

Wallrun Advanced Velocity:

Single Player Only (for now)

When using the wallrun system in a non-replicated game, you can also use the “Advanced Velocity” option - This allows you to add a Z influence value to mimic gravity (Slight up wards motion then a downwards motion) as well as a speed decay rate to force **“Fall off”** using the values **“Speed Decay Rate”** for per-tick decay rate of speed - **“Gravity Influence”** this will add then subtract the value from the velocity to mimic gravity impact - And **“Velocity to Detach from Wall”** will trigger a stop of wallrunning and enter to a falling state if the velocity reaches the defined value

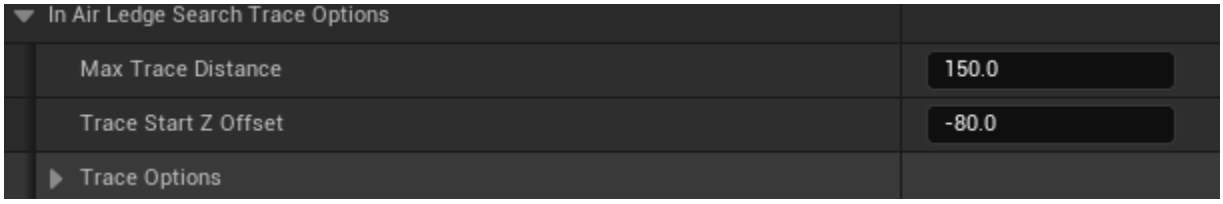


Note: If Velocity to detach is 0 - Wallrun will continue until object blocked, or canceled via input. To enable these settings just ensure to check the box named **“Use Advanced Wallrun Velocity”**

There is also an option to control how far the actor is placed from the wall during a wallrun to ensure correct distance at all times - **Use the “Distance to Wall”** value to influence how far from the wall the system will correct if enabled

Attach to Ledge From Air Settings:

After a jump you can attach to a ledge / climb point actor from mid air - The settings that control the distance a ledge can be found as well as the starting height offset are under **“In Air Ledge Search Trace Options”**



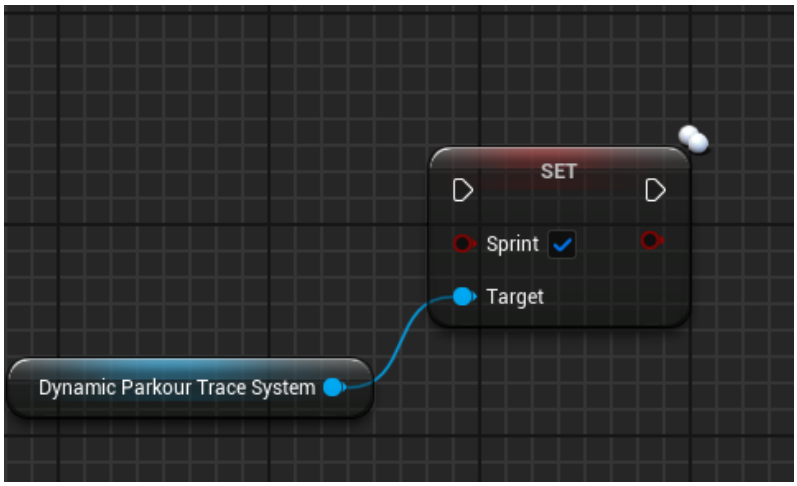
The Max trace distance controls how far the search for a ledge can go
The Trace start Z offset moves the initial trace start from the actor position to ensure that it can find ledges below the actor and verify its validity before attaching.

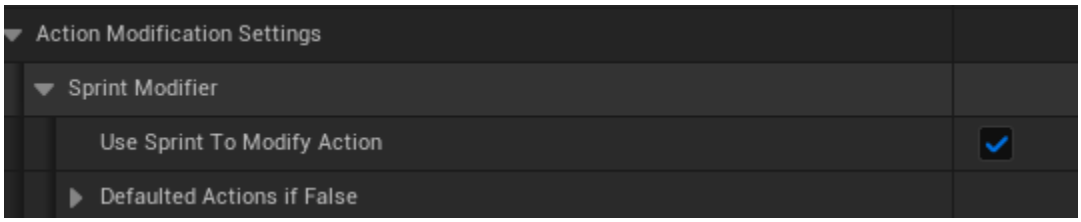
This also comes with the “Trace Options” menu to ensure full control over in air ledge detection range

Action Modifications:

Sprint Modifier:

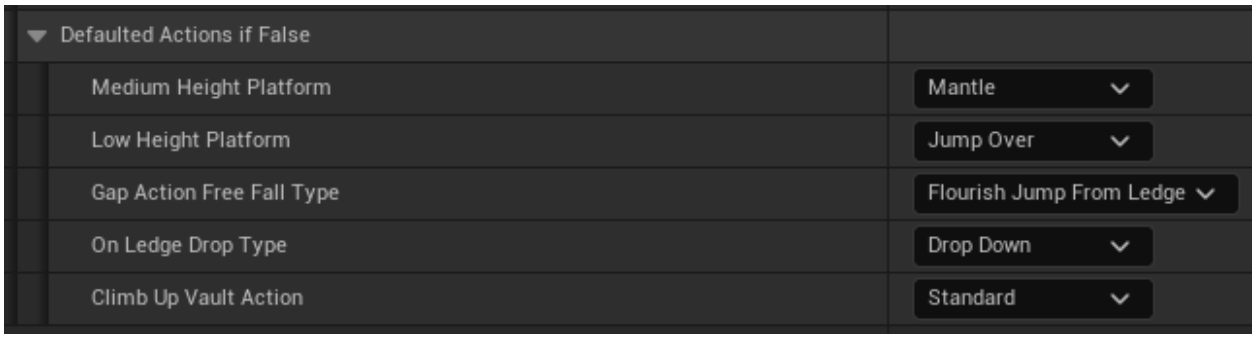
System has a built-in “Sprint” boolean that can be used in combination with the **“Sprint Modifier”** settings under **“Action Modification Settings”**.





By setting the **“Use Sprint to Modify Action”** to true - and adding the sprint boolean to your sprinting code the system will then override certain action defaults from an onto version to an over type instead for medium height and low height type actions. There are settings in **“Ground Action Settings”** under **“Vault Action Settings”** and **“Low Jump Action Settings”** to fine tune this override distance to ensure that onto actions are taken when the object is too long

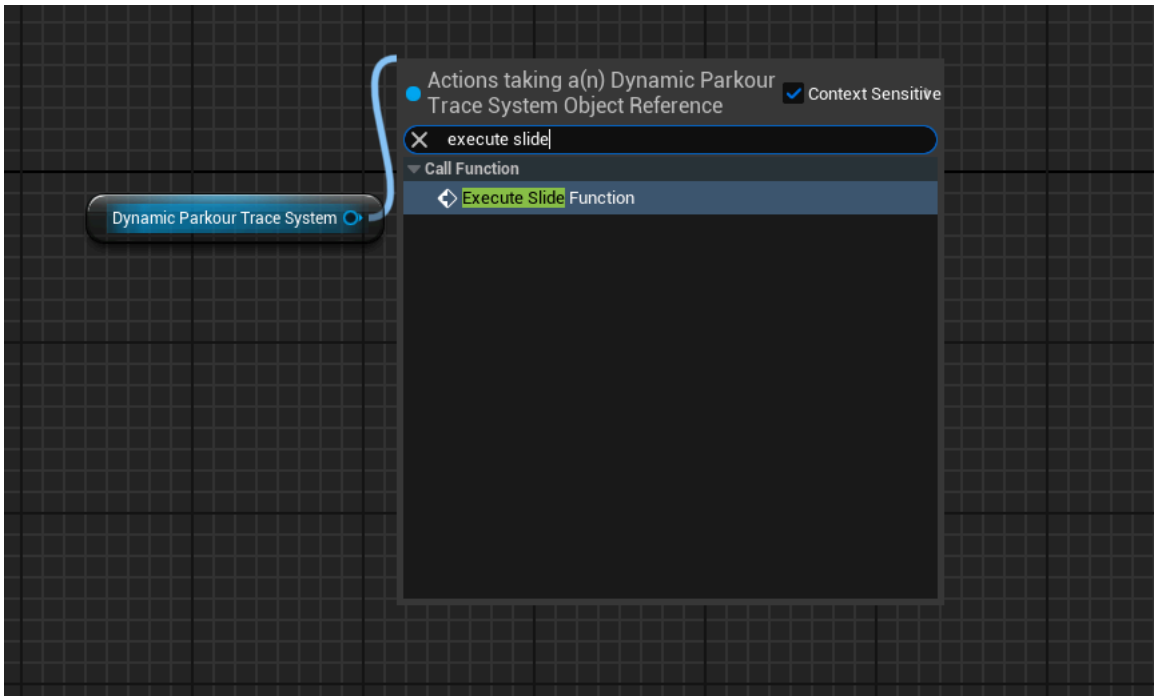
If you set the value of **“Use Sprint to Modify Action”** to false there is a sub-menu for selecting the default action choice - Ensuring that full control over actions are in your hands.

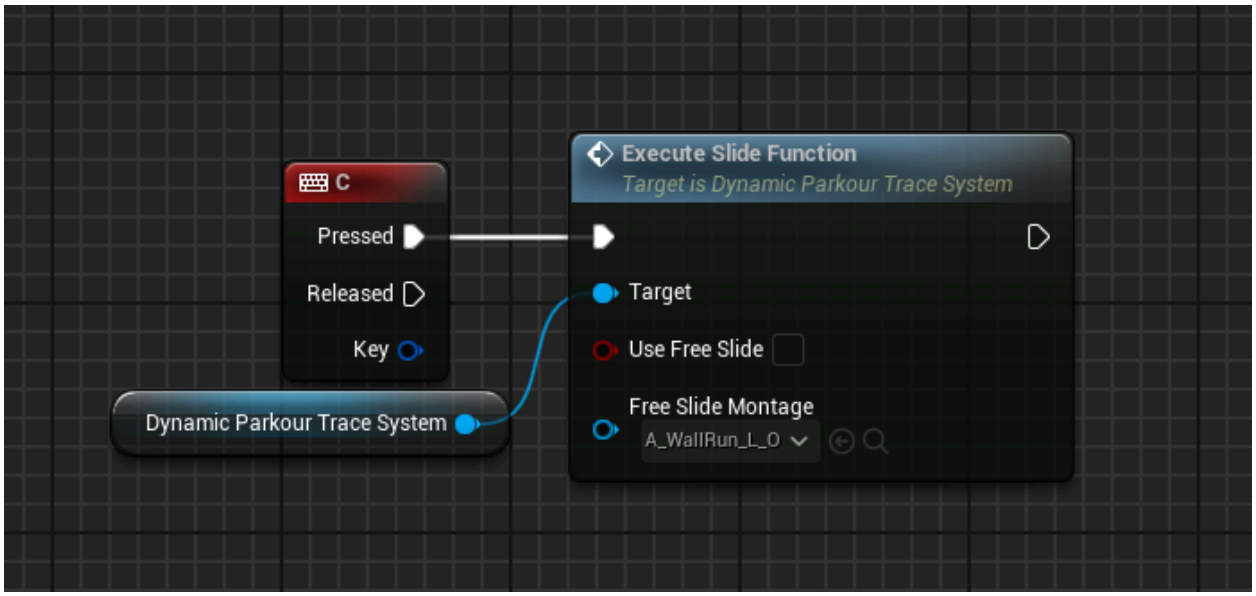


NOTE: When using a defaulted action - Ensure that this action is enabled in its respective menu. If an action is selected to be allowed by a modification (E.G Vault Over Long Obstacle chosen instead of Medium Height Mantle when sprinting) both actions must be allowed - If using defaults only the select default action needs to be set to allow - For a full list of actions affected by the various modifiers see the section marked “Action Override Table”

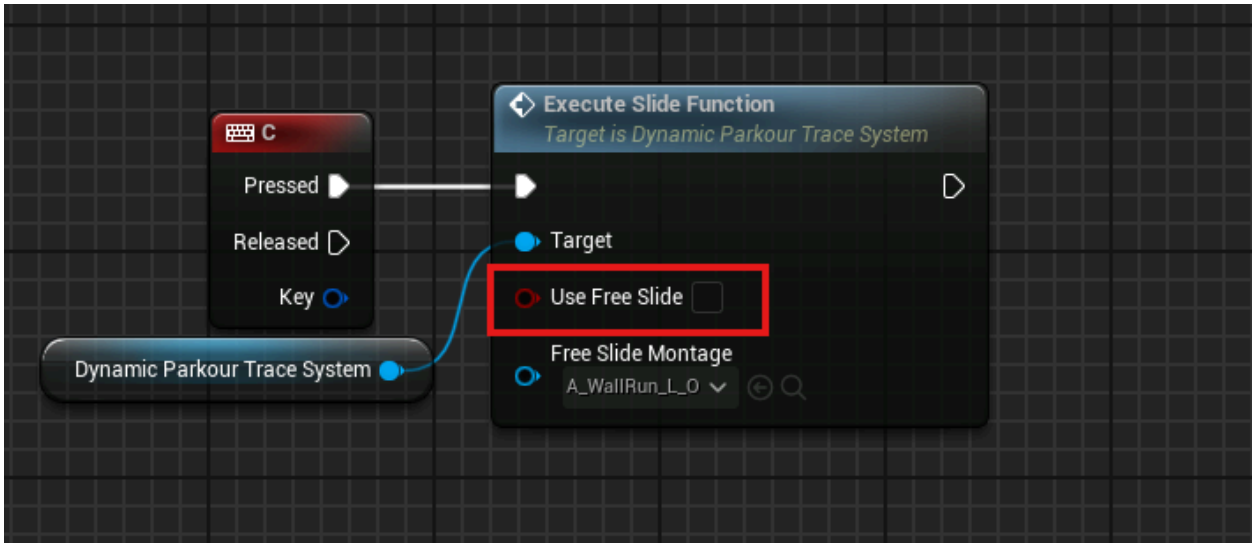
Slide Modification:

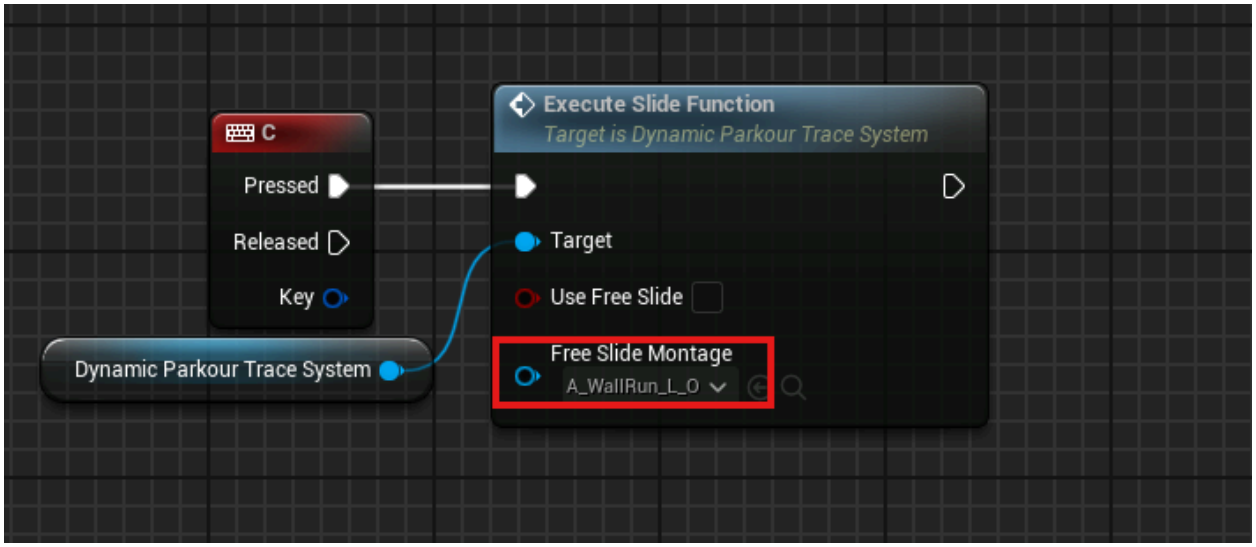
Sliding is separated into two types of actions using the built in **“Execute Slide Function”** event that can be called in the player BP using a reference to the **“Dynamic Parkour Trace System”** component, pulling off and typing in **“Execute Slide Function”** and hooking it up to your desired input





You can choose to use “Free Slide” where there will be no influence from the system on how slide will be chosen by checking the box that says **“Use Free Slide”** and assigning your slide montage to the montage slot provided in the function (no motion warping will be applied to this montage)





Or you can choose to use the system to decide if slide is possible using its trace based detection and providing back a useable motion warp location to slide under the midpoint of an object if its within the settings defined in the “Slide Area Dimensions Options”

If using the system to slide you can also use this input to influence action choice for gap action Up / Across and Downward action types as well as sliding or vaulting over if not height blocked depending if **“Primary Action Key”** was pressed or **“Slide Function”** was executed. You can enable this setting under “Slide Modifer” and check the box named “Use Slide Function to Influence Action”

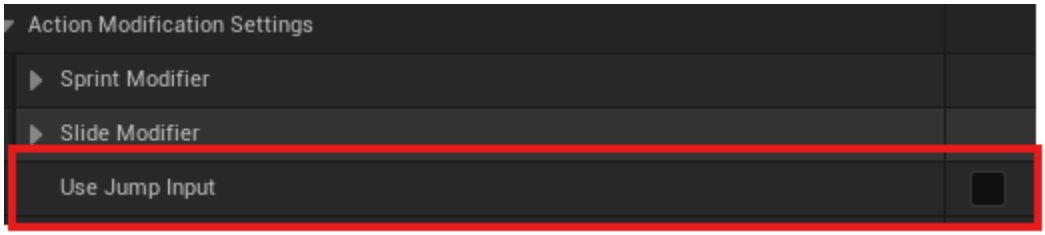
If desired to use system slide but do not wish to use the input to influence the action result of other action types, keep **“Free Slide”** to false in the **“Execute Slide Function”** and expand the menu “Defaulted Actions if Fase”

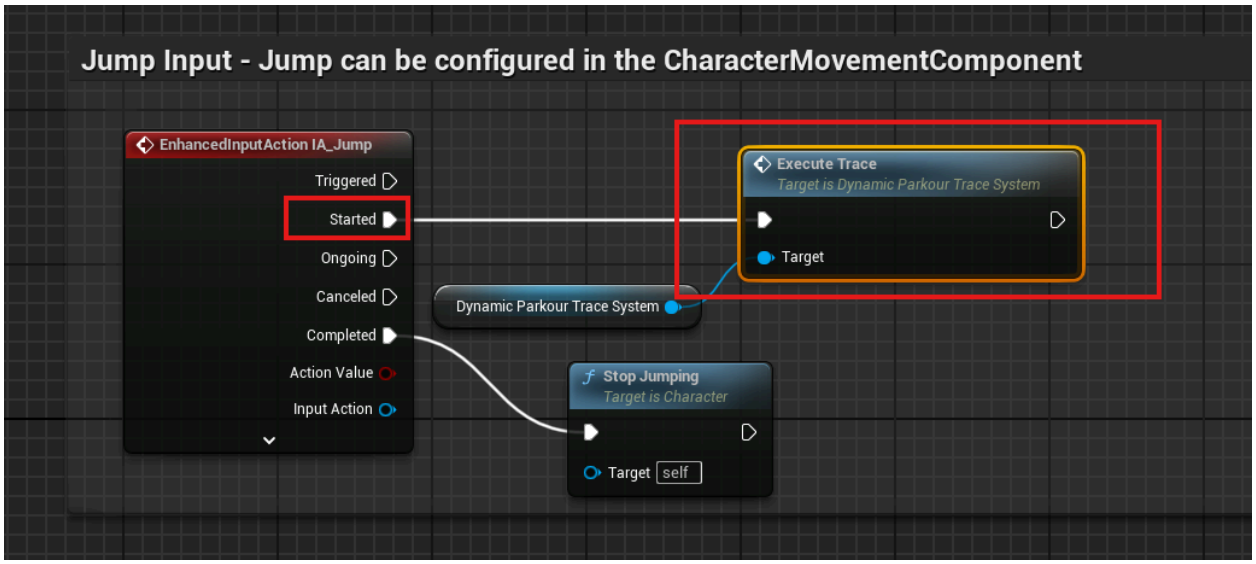
The **“Default Action Choice”** will only affect actions if both downwards and across actions have been detected.

Note: As with the **“Sprint Modifier”** action choice, all actions influenced by the modification will need to be set to allow, if using slide to influence. If not, only defaulted actions will need to be set to allow.

Use Jump Input:

If wanting to use jump to choose an action or a default jump, connect the **“Execute Trace”** function into your jump execution and REMOVE “Start Jump” From the execution and ensure to mark the option **“Use Jump Input”** under **“Action Modification Settings”** to true



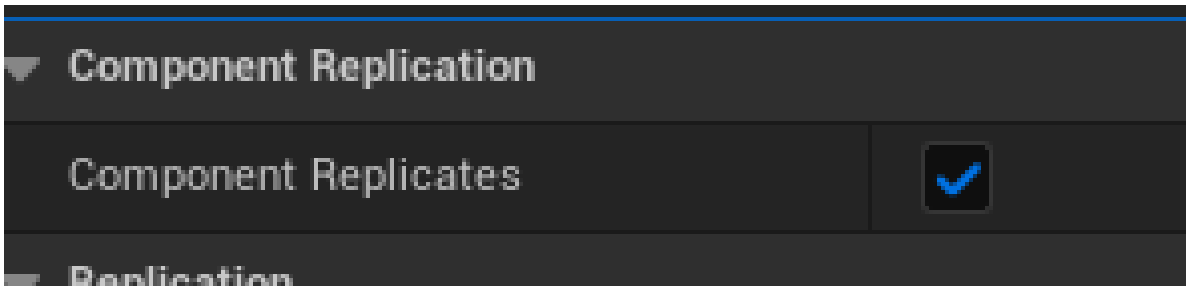


Note: Execution changed to started vs triggered to stop any possible double calls - Does not affect jump as the function for jumping is called internally back to the character to ensure that jump still works (Just with some additional flair)

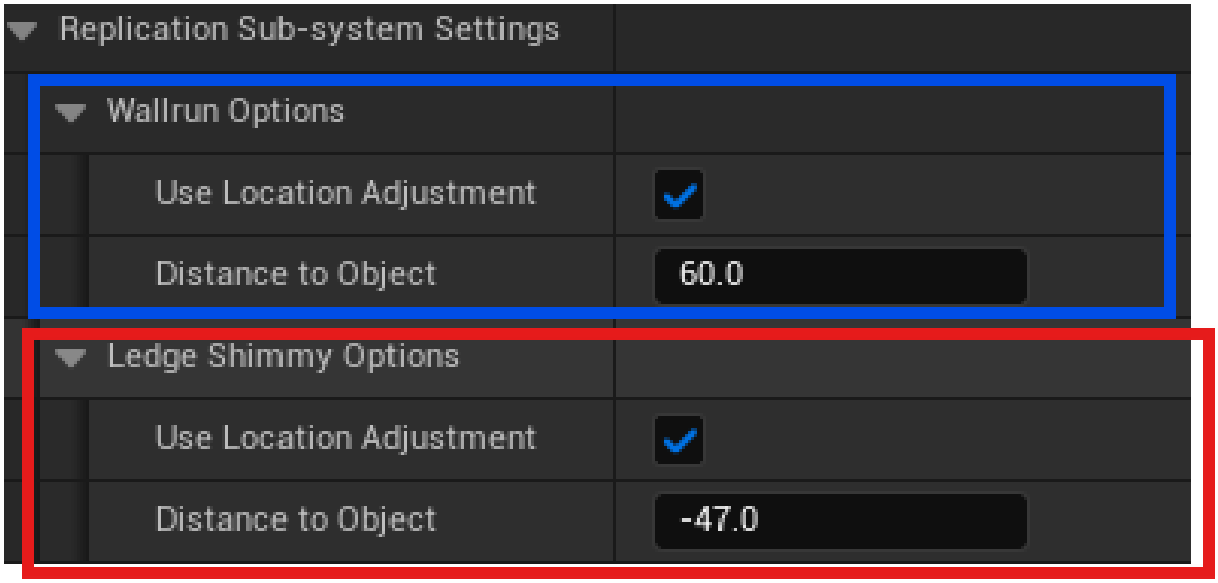
Also for those who have been users of Version 1 of this system - The previous call for checking if “Busy” is true or false is now handled internally and is not needed anymore

Replication & Replication Overrides

Replication is handled completely internally by the system using internal RPC calls and server events when necessary - These events are ONLY called if you set the Component to Replicate by clicking on the **“Dynamic parkour Trace System”** component in your components list of your character, then in the details panel search **“Replicate”** and ensure that **“Component Replicates”** is set to true



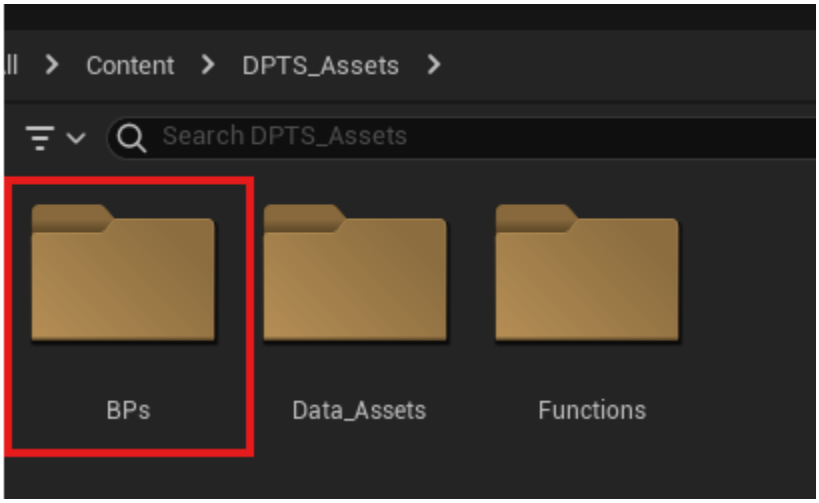
Certain sub-systems are not currently able to be fully replicated, however there are options for single fire events to provide the Location Adjustment aspect for climbing and Wallrunning but in single execution events rather than checks with smooth interp for location and rotation correction. To enable these expand the menu **“Replication Sub-system Settings”** and set the values of **“Use Location Adjustment”** under **“Ledge Shimmy Options”** and **“Wallrun options”** with a value to set how close you wish to be from the object when this even fires to correct location (For ledge actions this fires when the notify **“Set_Climbing”** is called, and for wall running it is called after confirming the wallrun area is valid and applying rotation)



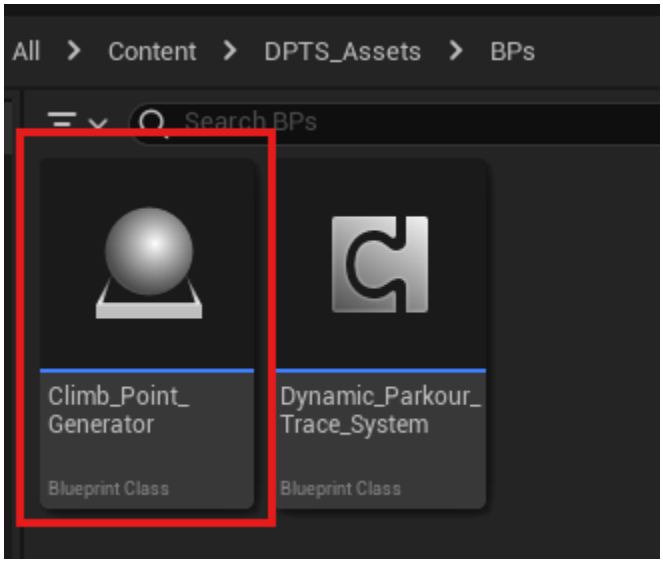
Climb Point Generator

Primary Settings / Initial Setup

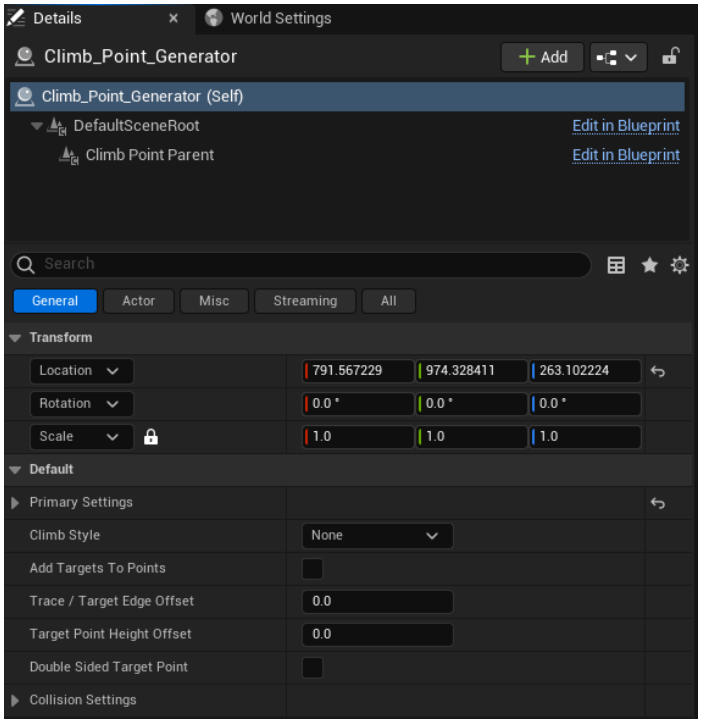
Included with the asset is a climb point generator BP with various function - To locate this blueprint, first open up the **“DPTS_Assets**” folder and double click on the folder named **“BP”**



Then inside you will find two blueprints - Once is the primary system component, the other is the climb point generator. To begin use just drag the **“Climb_Point_Generator”** blueprint into the world



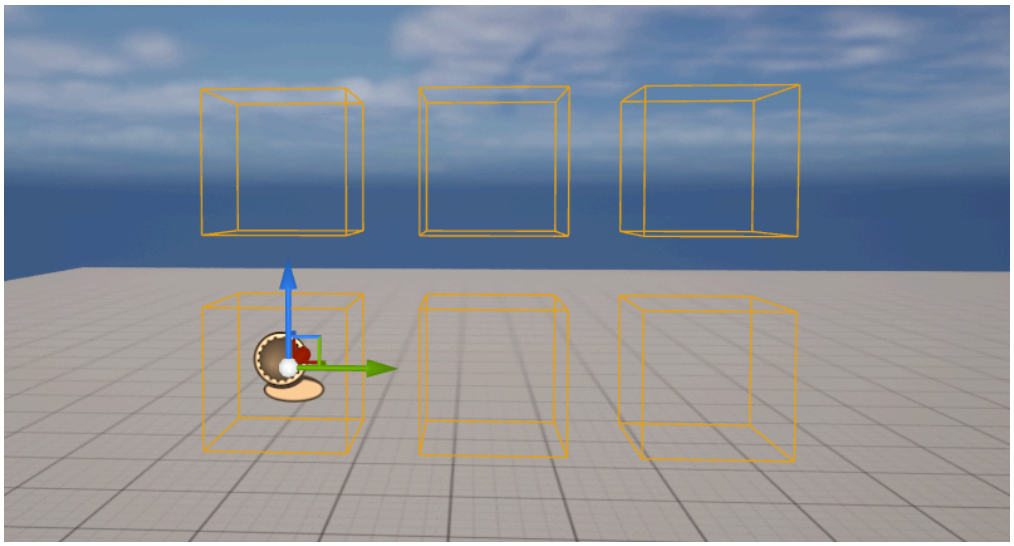
Once an instance of the generator has been placed, navigate to the details panel while the blueprint is selected and you will see a list of settings for the generator -



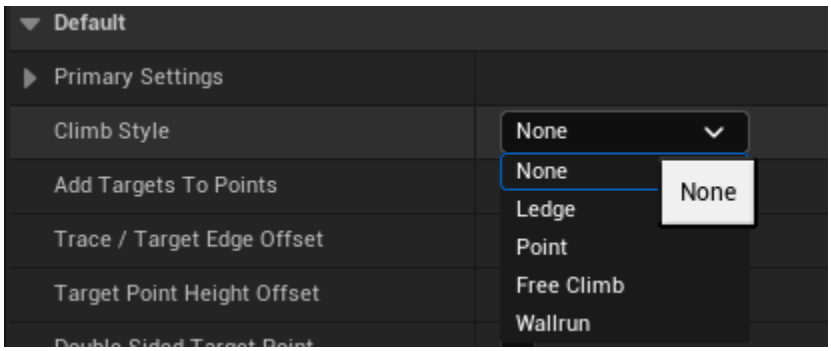
First expand the menu **“Primary Settings”**

The generator has the ability to spawn collision boxes in a single instance or a line (Using the **“Points Per Row”** Value, if set to 0 there will be no boxes spawned) as well as a grid fashion (Using the “**Points per Row Value”** as well as **“Number of Rows”** to create a grid pattern of climb points)
Use the **“Space Between Points”** value to add distance between spawned points (Distance added is even between all spawned points)
The **“Box Extent”** value is used to control the size of the spawned point (Note: Box extent is relative to actor overall rotation, rotating actor does NOT affect box size)
The X value is best used for thickness, and the Y value for length when sizing your points (specially if using target points)

Once you have adjusted the primary settings to your liking you should see something like this depending on the settings you put in



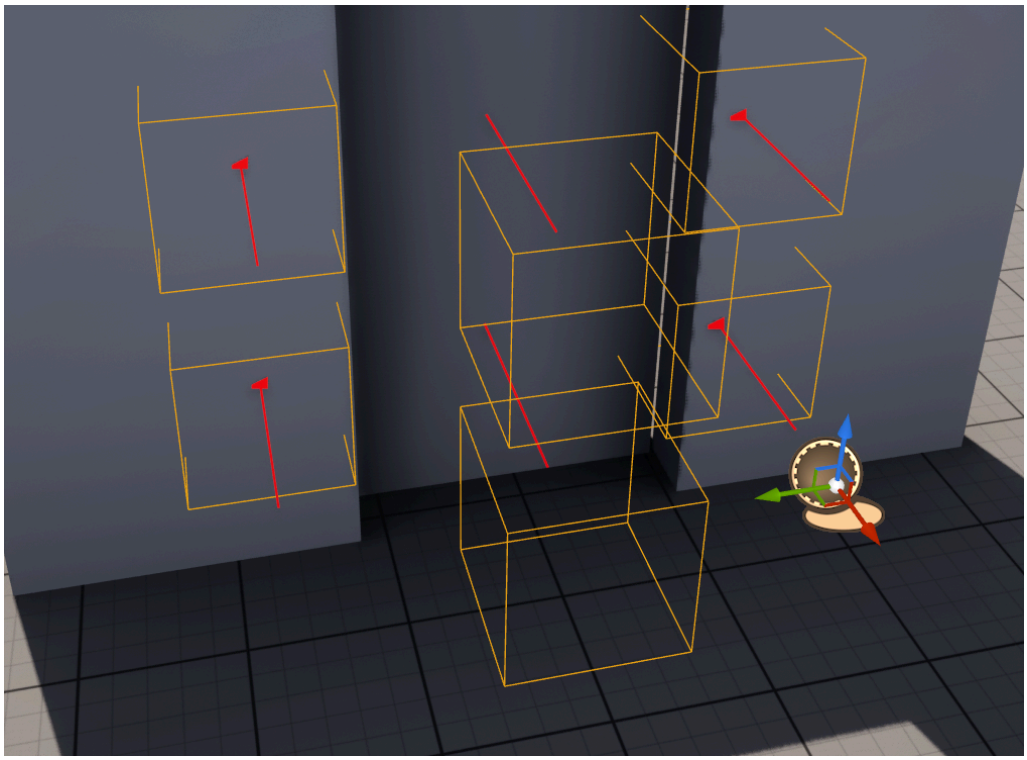
From here the only necessary action is to ensure that a climb style is selected for this climb point: To do this navigate down below the primary settings menu for the option listed as **“Climb Style”** and select the desired climb style that this spawned generator is meant to be



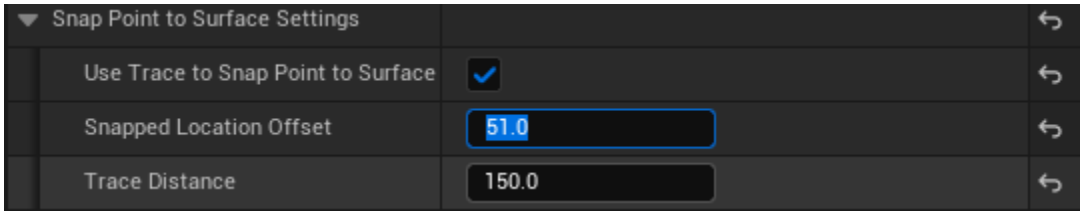
Snap Point to Surface:

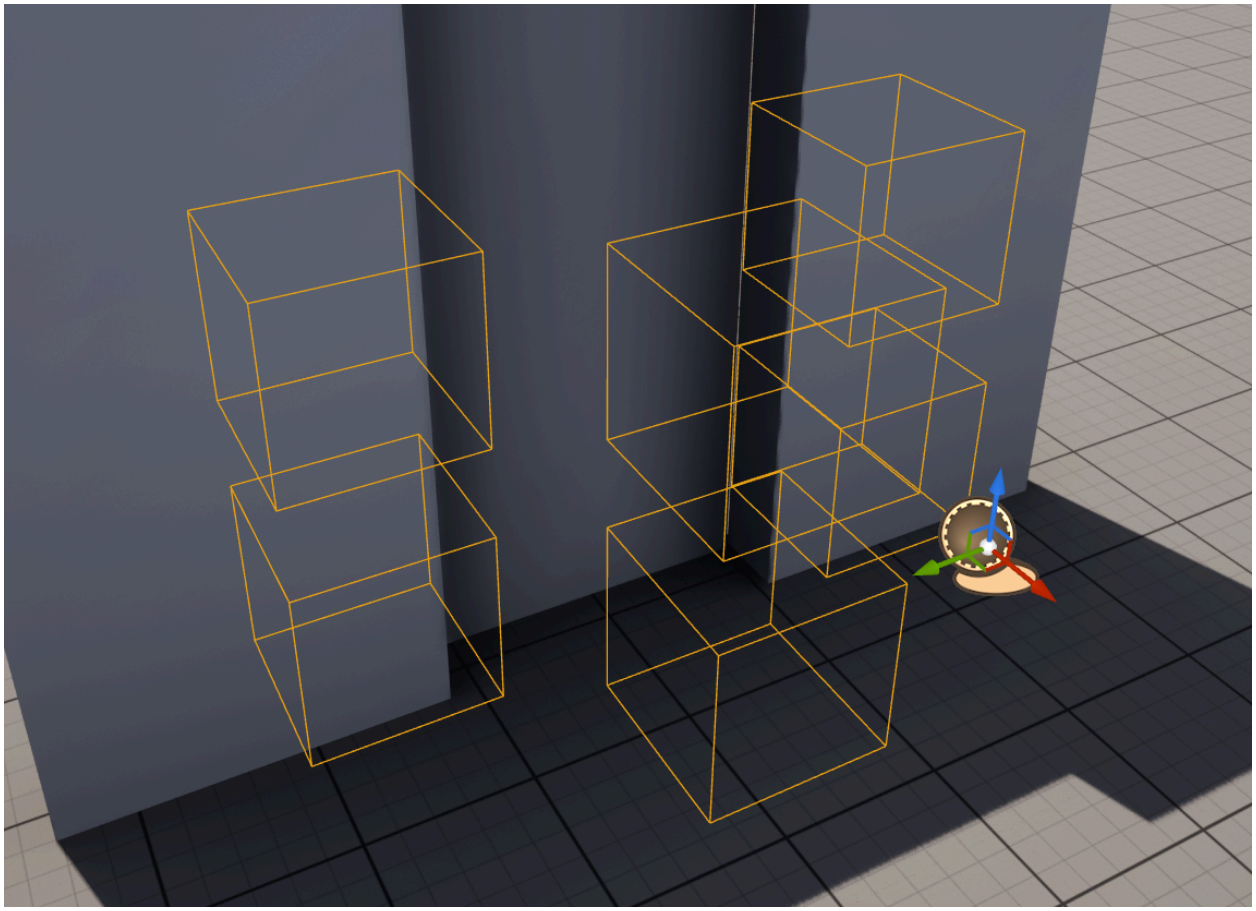
If trying to place on a wall that is uneven and wish to ensure that the points attach themselves to the wall correctly, open the menu **“Snap Point to Surface Settings”** and check the box that says **“Use Trace to Snap Point to Surface”** and set **“Trace Distance”** for how far back you wish to have the trace run to find a surface

Once enabled and a trace distance has been set, you will see a trace run when you move the blueprint, move / rotate until the trace is in line with the surface you want to pin the points to and move until trace detects the surface - Points will automatically snap to the found surface object



Offsets can be added to the locations using the **“Snapped Location Offset”** value - Adjusting this will help correct point position while maintaining the snapped location

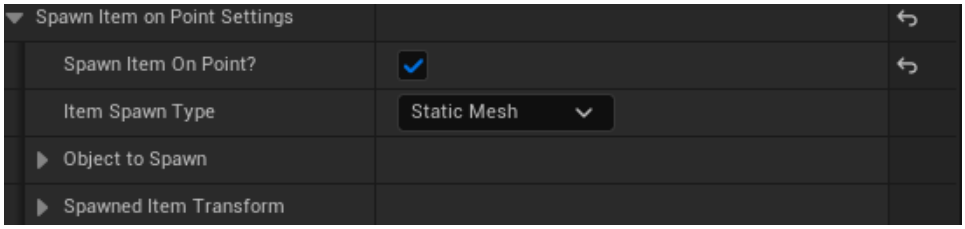




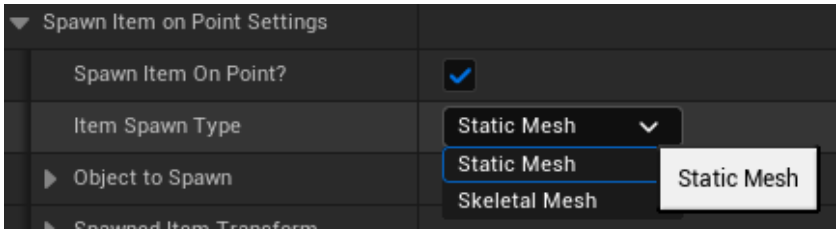
Note: Value **“Box Extent”** X axis value changes where the trace starts ensuring that the trace always starts from the points end rather than actor position - This X value is also a good number to use for Location offset to have the box sit on top of the surface.

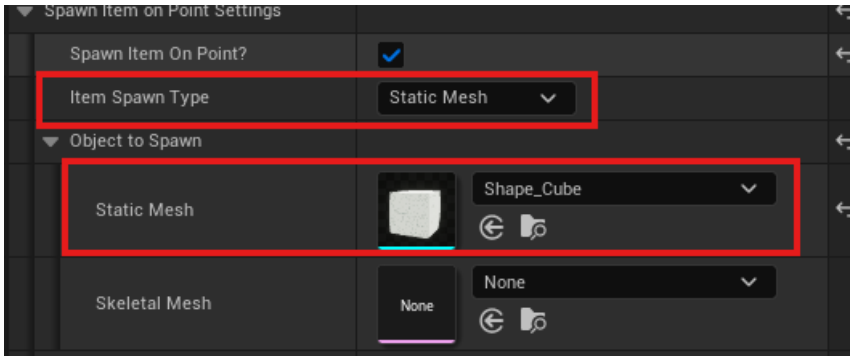
Spawn Item On Point:

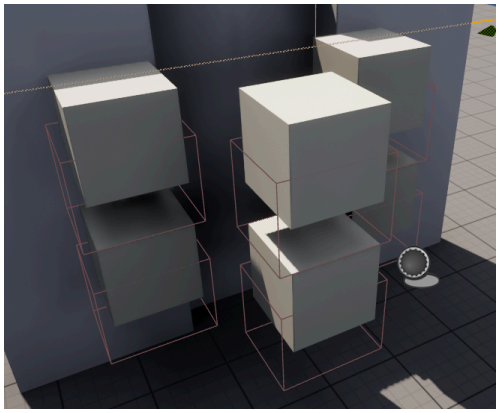
When spawning climb points you can also add static and skeletal meshes to each point, allowing easy set up for climb additions into your map. To enable this open the menu “Spawn Item On Point Settings”



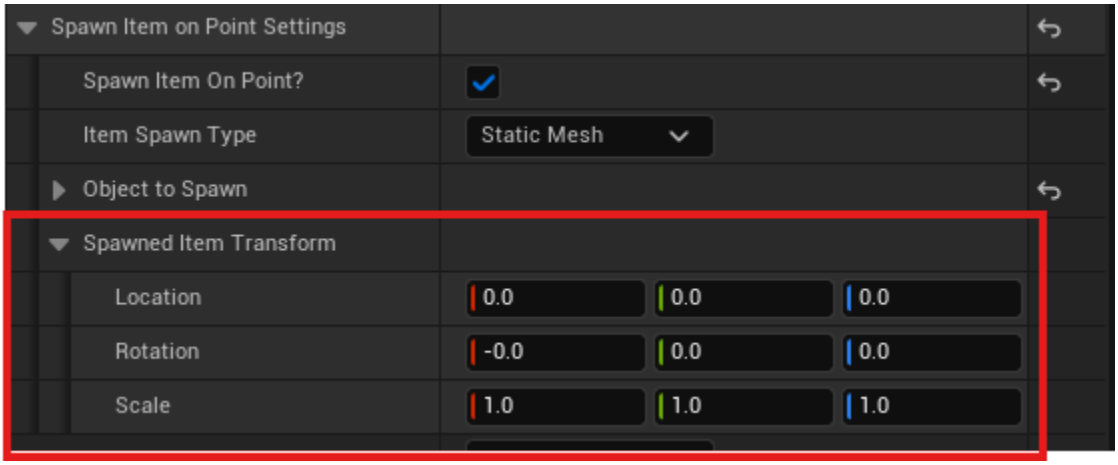
You can choose between a **static mesh to add or a skeletal mesh** and then select the specific object of that type you wish to spawn by expanding the Object To Spawn Menu and select the object through one of two available menus



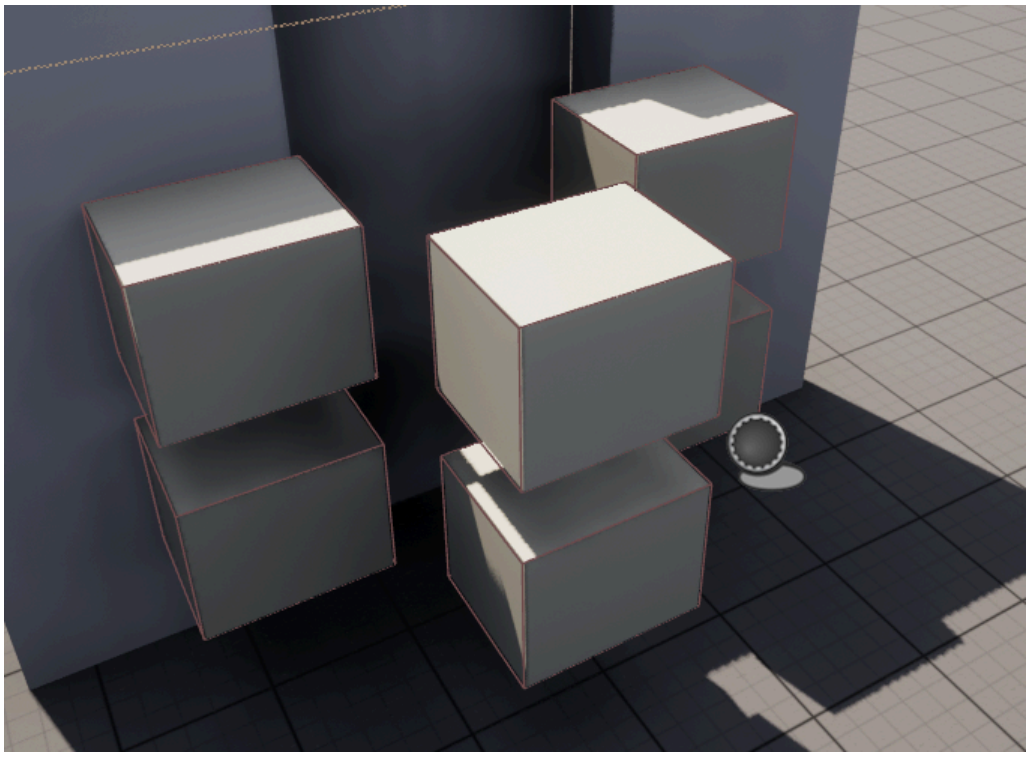




You can add any object type of skeletal or static mesh that you wish - Once added you can also use an additional set of settings to control the spawned objects transform by opening up the **“Spawned Item Transform”** menu



You can use this to provide location rotation and scale values to better fit your needs to fit within the points area

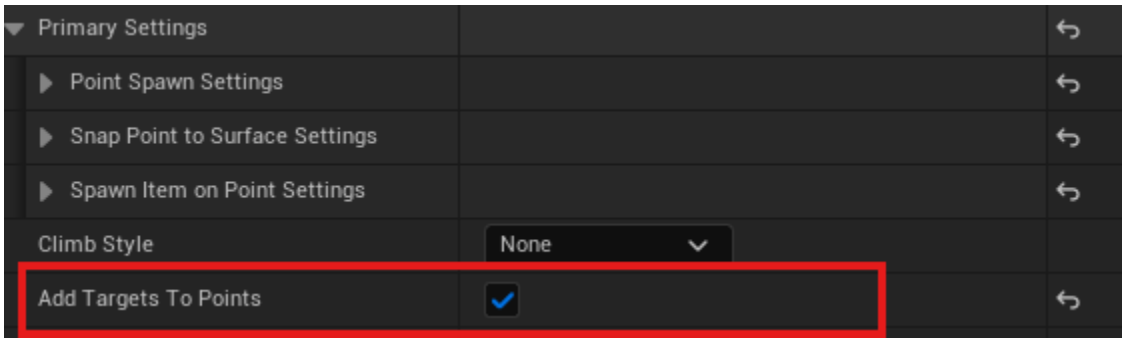


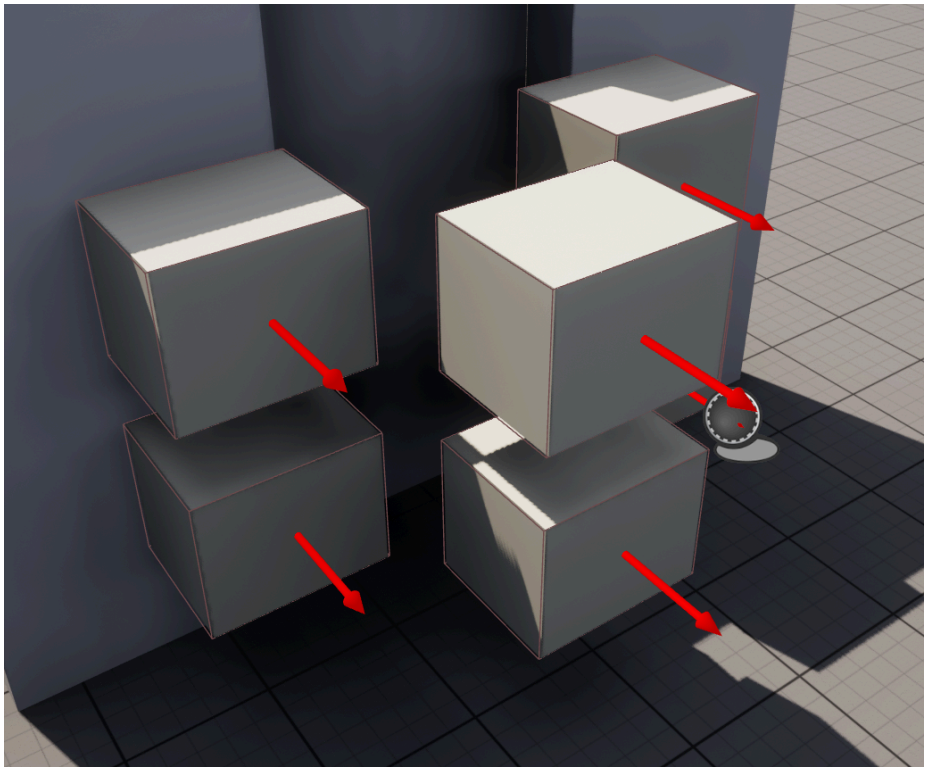
Point Target Settings:

Climb Points have two ways of feeding back locations - First is through the primary systems traces to locate the objects ledge
The second is through an added point to each climb point called a “Target Point”

This target is represented by an arrow on each spawned Climb Point when enabled

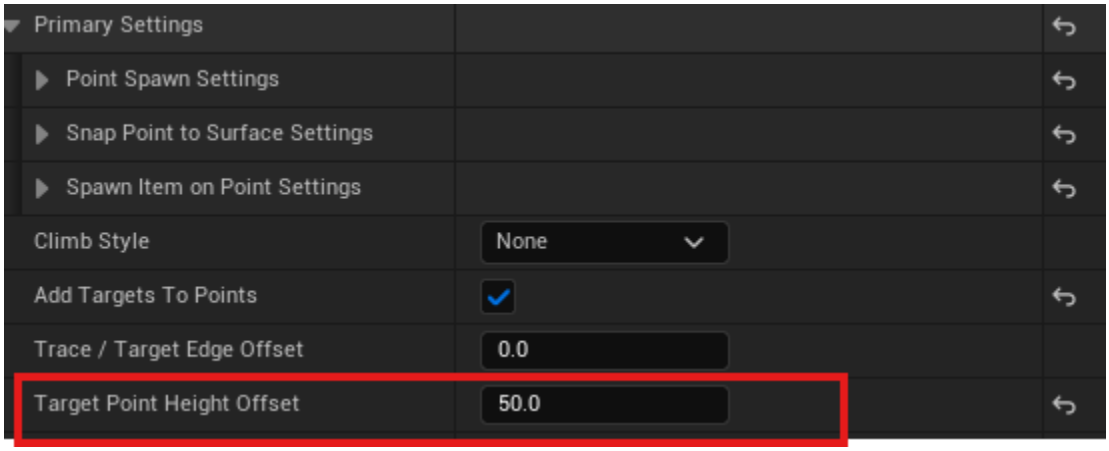
To enable this option just ensure the box is checked that is named “Add Targets To Points” and you will see an arrow appear in front of each point

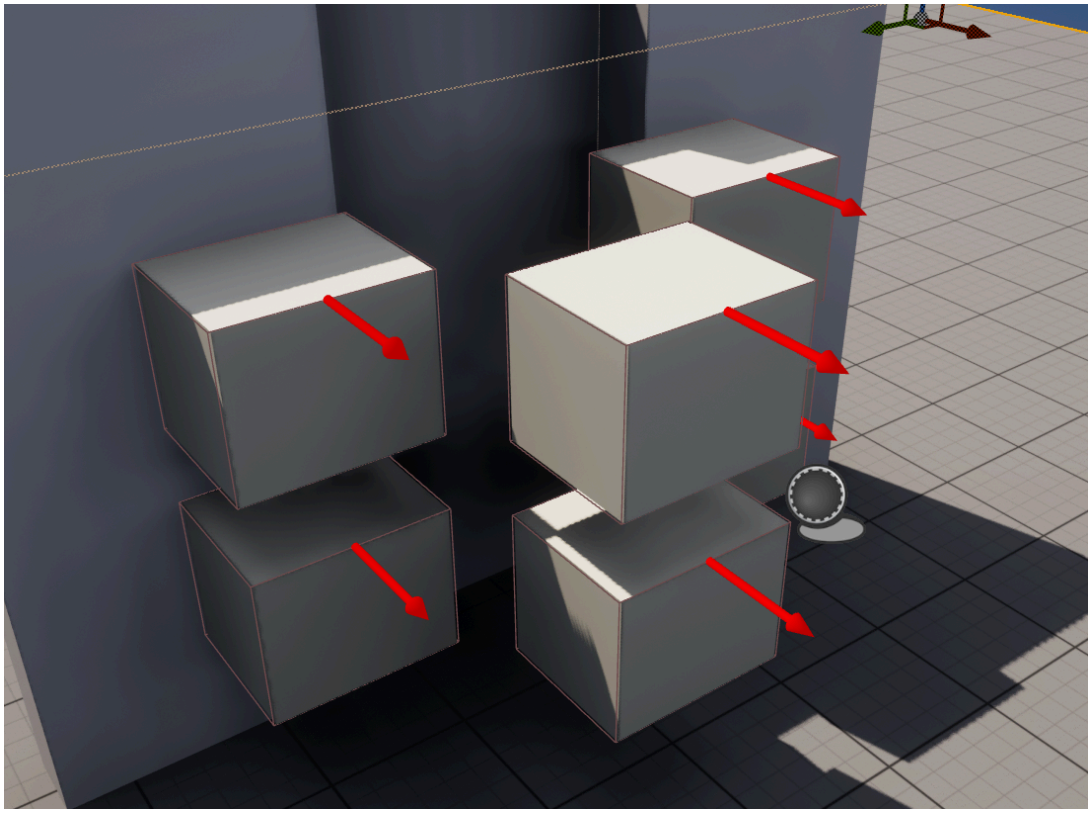




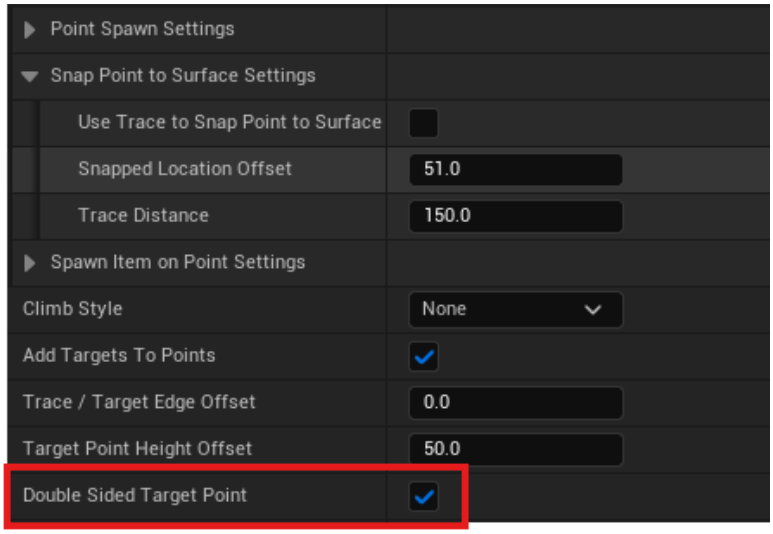
These target points can be adjusted using the two settings **“Trace / Target Edge Offset”** and **“Target Point Height Offset”** (**NOTE**: **“Trace / Target Edge offset”** affects target point AND trace for surface - Primary use is if you need to modify the point target locations so heavily that the points and / or traces no longer line up with the box end and need to adjust its forward / backward location)

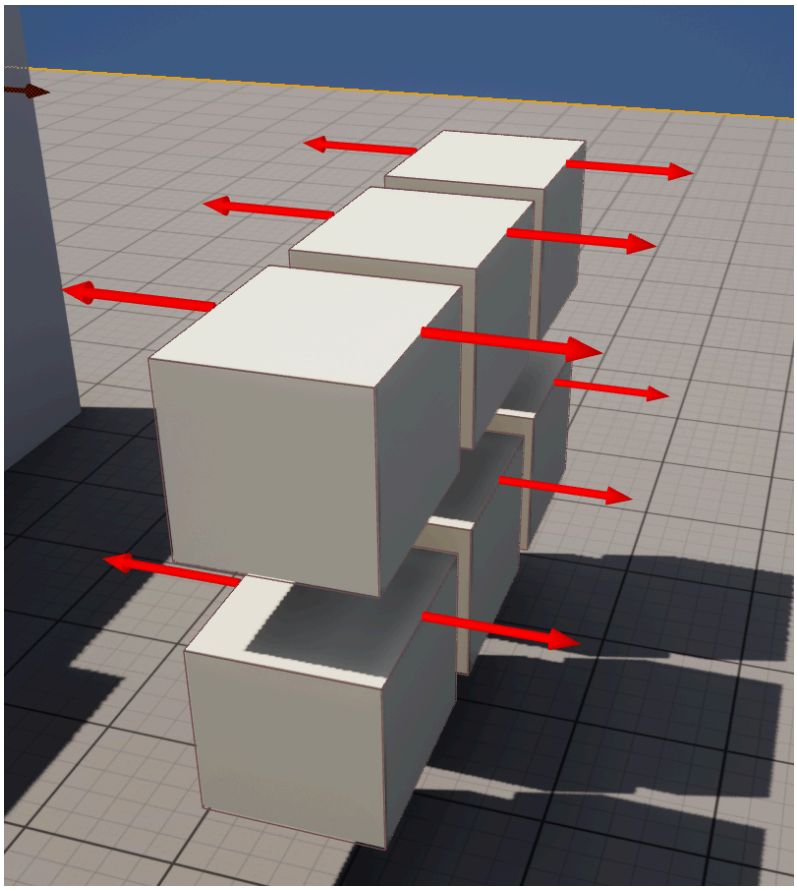
For most cases you will only need to play around with the **“Target Point Height Offset”**





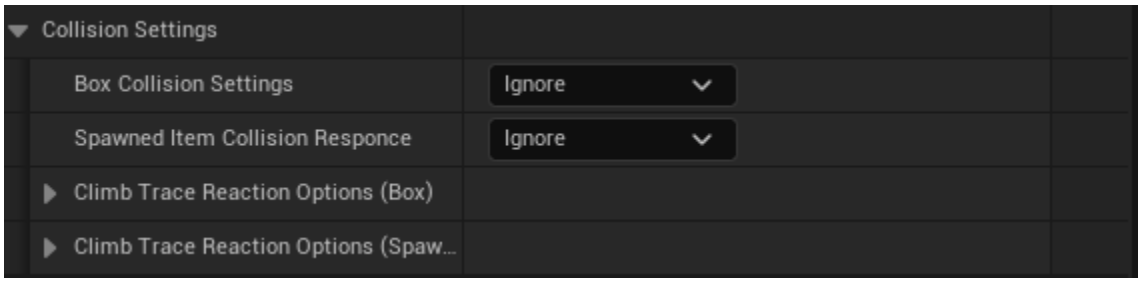
In cases that you wish to use target points and have climb points that can be used from both sides - There is an option you can enable **“Double Sided Target Point”** to spawn targets on two opposite sides of the climb point



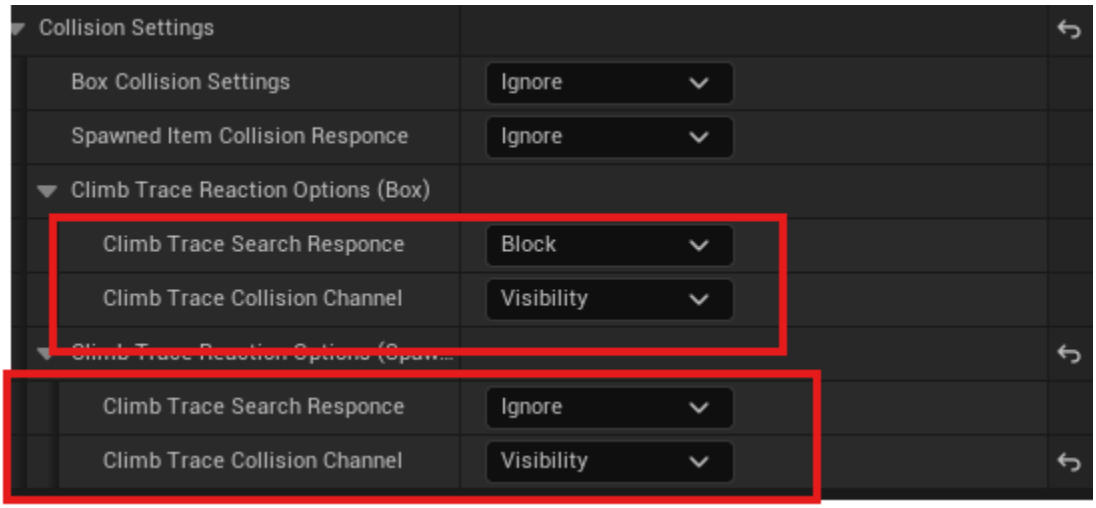


Collision Settings:

Collision for the spawned in box collision as well as any spawned items can be adjusted in the **“Collision Settings Menu”** and set the response to all channels allowing for it to block, overlap, or ignore everything in the world around it



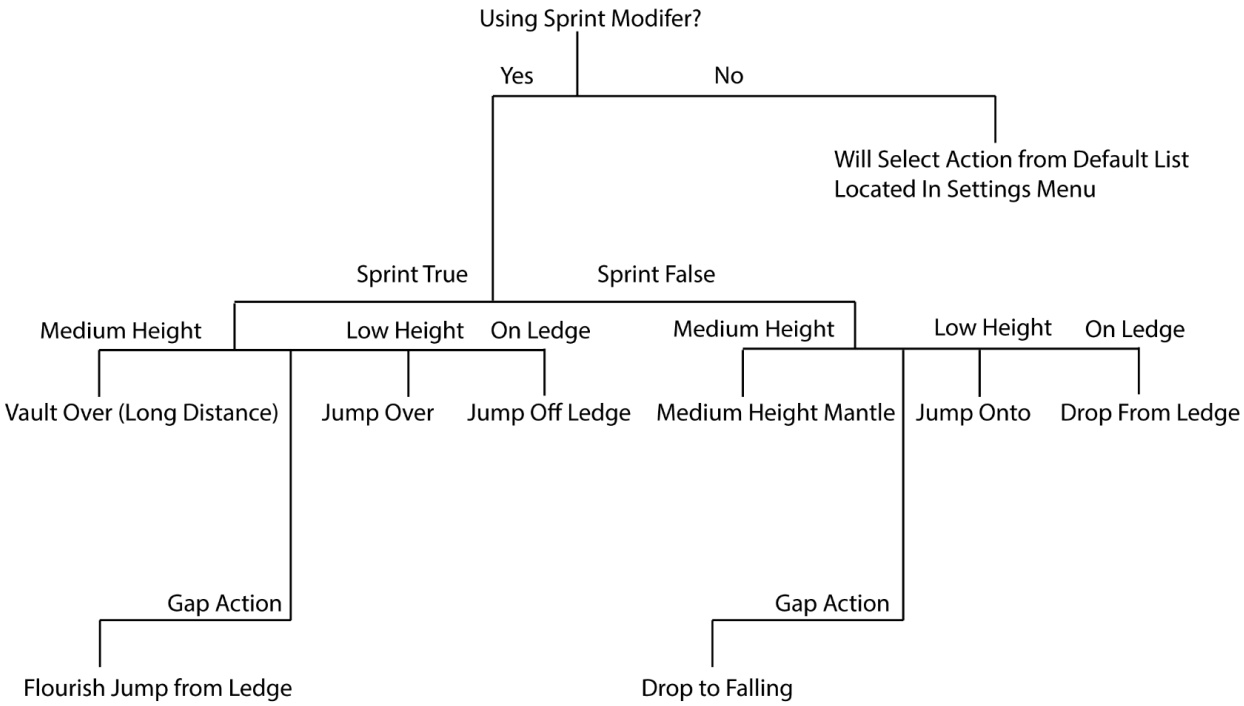
To ensure that an objects collision doesn't impact its ability to be used as a climb point there is a secondary menu for both **Spawned Items** and **Box Collisions** allowing you to set the trace channel being used by the climbing trace in the **Dynamic Parkour Trace System** component settings, and the specific spawned items response to said channel



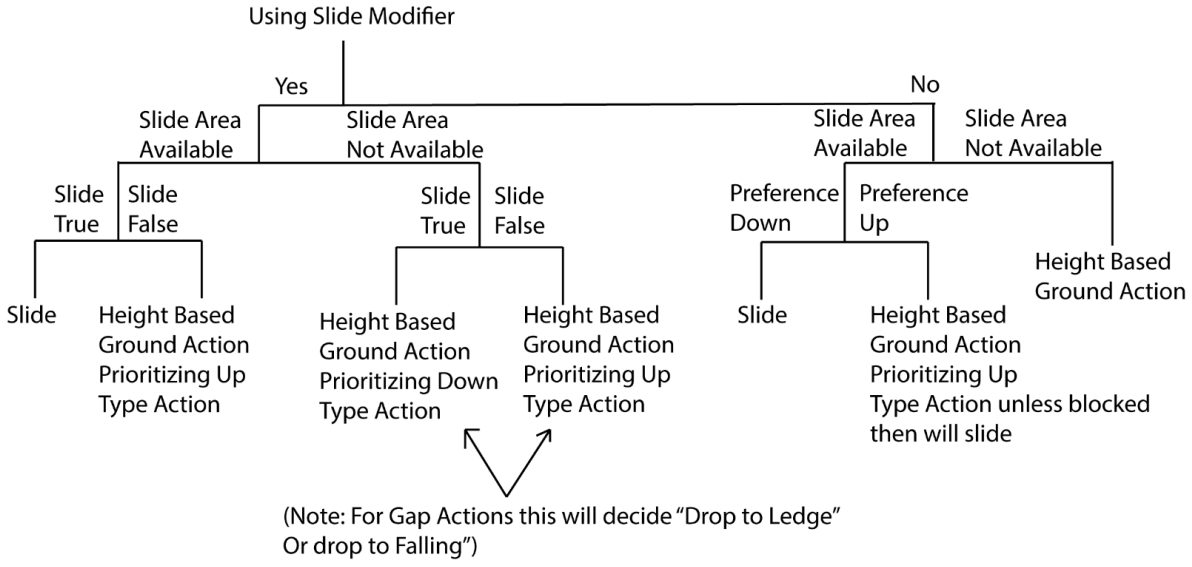
This allows you full control over what the system will trace, and how the collision with the actor impacts the world in game.

Actions Modifications Chart

Sprint Modification Action Override chart for Platform type obstacles / Climbing



Slide Modification Action Override chart



System Patch Notes:

Update: V2.05 (11/18/24)

Primary Bug Fixes

- Corrected drop action from taking drop to ledge or drop to falling incorrectly if action was not set to allow
- Corrected On ledge Vault over and Vault 180 to opposite ledge action selection if actions are not set to allow
- Adjusted ledge detection mechanics to provide system ability to find ledges closer to the wall areas

System Changes:

- Added new menu “Advanced Modifications” with sub menus for “Ledge Check Advanced Settings” and “Climbing Trace Advanced Options” to allow for further defining of ledge objects for system to use for climbing
- Changed behavior when not using modifiers for gap actions “Drop to Ledge” and “Drop To Falling” selection: Now if both actions are set to allow and not using modifiers, if no found platform is below the player (Defined by the trace settings it gap actions “Downwards Action Trace” “Trace Settings”
- Added variable descriptions to system settings to help better understand what each value changes within the overall system and added a large description of action warp location representation to the warp points in each action offset menu.

Update: V2.04 (8/25/24)

Primary Bug Fixes:

- Fixed wallrun only working for left side - Will now work left and right as intended

- Corrected Ledge Corner Transitions warp locations to ensure no unnecessary drop in Z location occurs

- Fixed **"Jump Behind to Ledge"** action not being influenced by **"Max Jump to Distance"**

System Changes:

- Re-created trace for **"Attach to Ledge (In Air)"** action to increase trace accuracy and action consistency

- Refined certain trace values to allow for climb actors to be closer to walls and not be accidentally ignored (If using complex advanced collisions, see climbing debugging video for further details on how to fine tune this even further

- Added a new drop down menu to the **"Reset Parkour Action"** notify to allow you to now also set the type of movement mode on action end, helping smooth out some rougher transitions from falling actions or actions that end in a stand / run position

- Added colorized comments in certain areas of key functions and code to better assist with understanding what values can be used to fine tune the system even further (More will be added in the future)

Update: V2.03 (8/11/24)

Primary Bug Fixes:

- Jump loop occurring when near an object and **“Attach to Ledge From Air”** disabled but **“Use Jump Input**” enabled

- Print string trigger when landed has been removed (Accidental leave behind from testing)

System Changes:

- Stopped non-wallrun actions from occurring on climb points set to **“Wallrun”** climb types (Only Wallrun and Wallrun jump actions will occur on these types of climb points)

- Using slide modifier no longer also triggers a jump action if no object found and pressing the input executing the slide function and “Free Slide” set to false

Update V2.0.2 (8/05/2024)

Primary Bug Fixes:

- Certain action traces still appearing regardless of draw debug type set in respective menus

- Wall Run Jump Action is now being properly selected (Override was happening within the system under certain conditions and has been resolved)

System Additions / Changes:

- Added Collision back to “Tic Tac Jump” type action and Wall Run Jump Actions to ensure that when these actions are chosen the player will still collide with the world to avoid clipping / pass through

- Tic Tac Actions can now be performed under any height so long as the height is within the range set in the “Tic Tac Action Settings” menu (Previously was being overridden if not a “High” type obstacle)

- Gap actions now influenced by the “Trace Z Length” under the “Gap Action Trace Settings” > “Up / Across” to ensure that gap actions do not attempt on low height differences (**Note**: This value should be played around with to ensure that it traces your map accordingly: If your terrain is very rough / jagged i recommend setting this value to something higher to avoid a trace not hitting due to a terrain height difference)

Update V2.0.1 (8/02/2024)

Primary Bug Fixes:

- Ledge Drop Action not finding climb point correctly
- Ledge corner actions not being influenced by settings value “Distance to ledge to perform action”
- Gap Actions setting “Distance to Ledge” sometimes overriding and using an action instead of jumping when using jump input
- Rotation values of Pitch and Roll being provided into the actor rotation influence (Now only uses yaw)

System Additions:

- Added a boolean value to the notify “Reset Parkour Action” that when checked will execute the in air trace allowing you to use certain actions end to chain combos easier (**Note**: In air trace runs after a .2 delay by default and allows you to keep the notify position and check the box and the trace will run just slightly after the animation ends)