Skydiving and Paragliding Ver 2 - Documentation

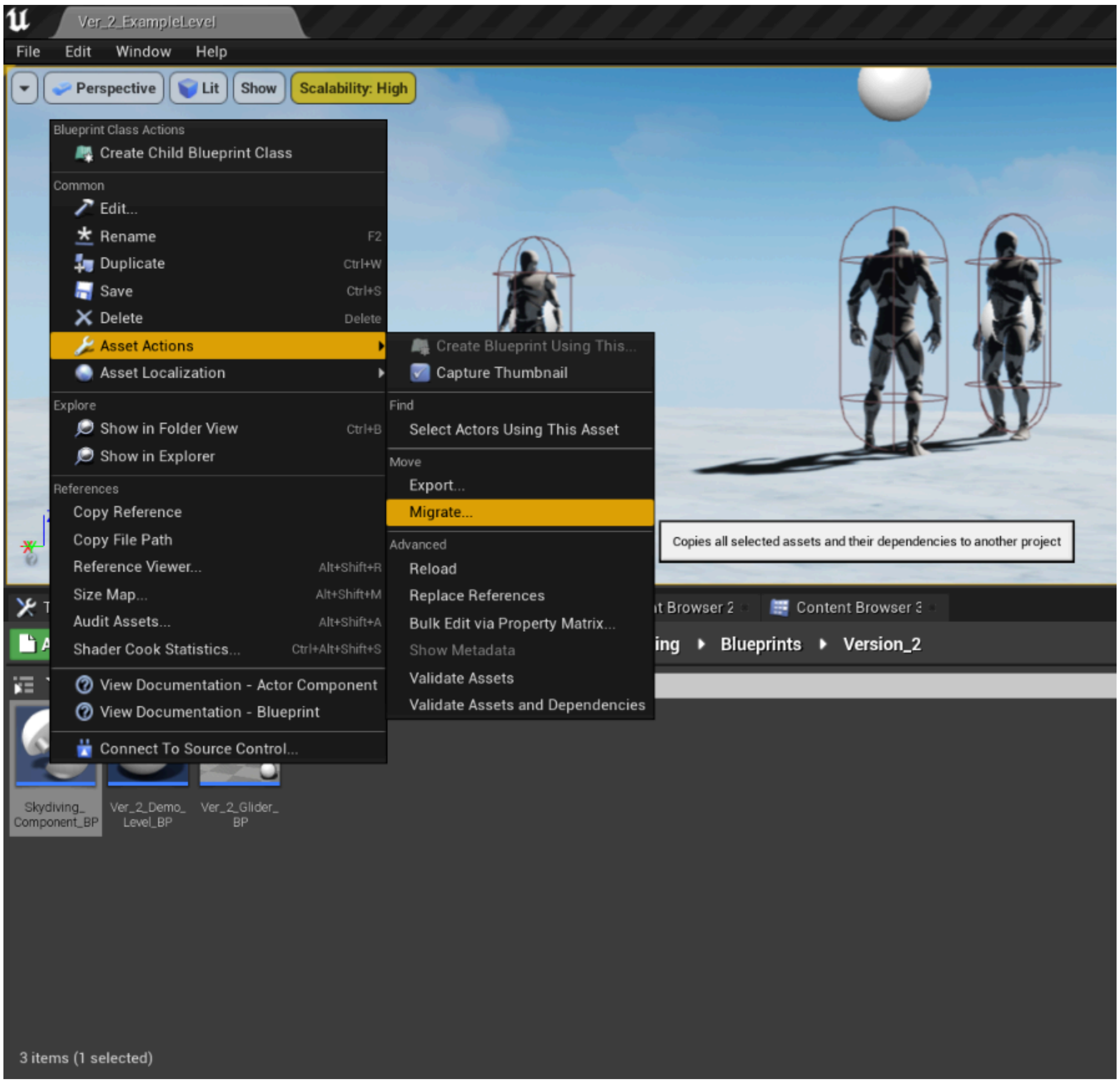
The skydiving and gliding animations can be downloaded here. You can test them with your Character before getting the project from the Marketplace. They can be used in your personal projects as well even if you don't buy them from the Marketplace, although they can't be resold as they aren't CC0. **The *rootbone* of the skydiving and gliding animations found in the downloadable link have modified location and rotation values, but the ones included in the project have fixed values (0,0,0).** Also, the gliding animations won't work as expected without the **Hand IK** used in the project. You can see in the Demo how they look without the IK.

The animations were made using the default Epic Skeleton. If you have a Character with a different Skeleton, both hands transform values may not retarget properly, that's why you can download them and test them beforehand.

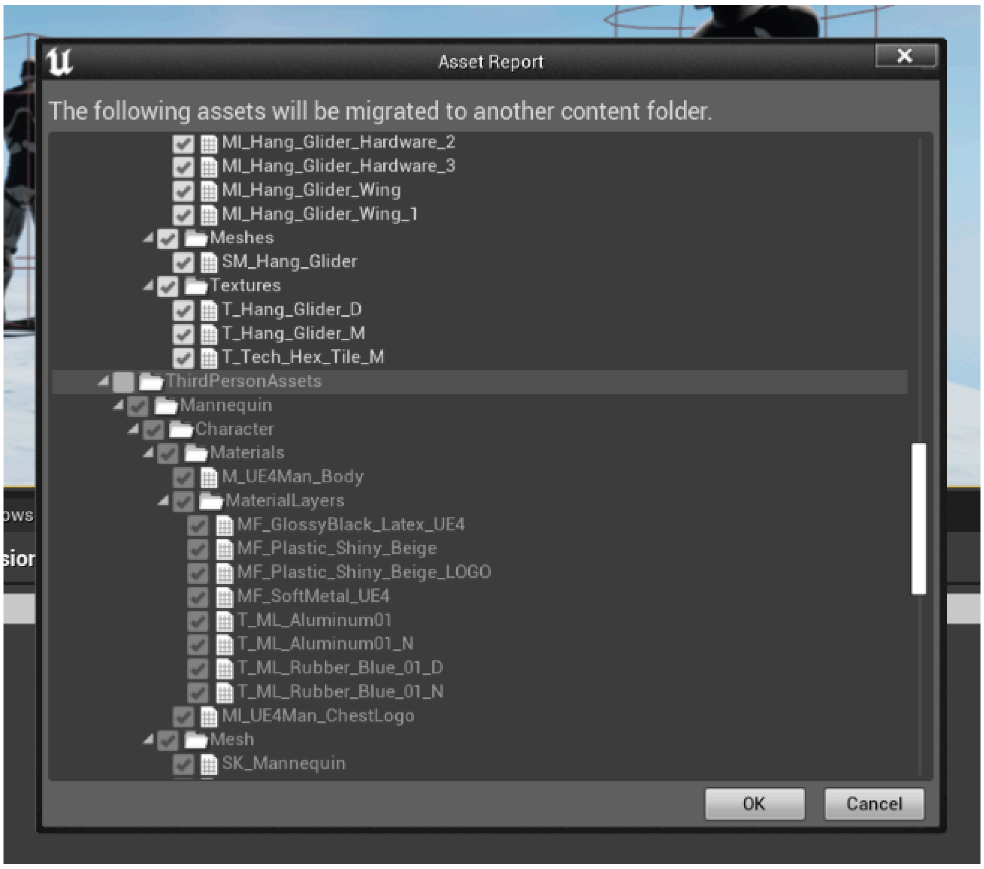
First test the Hands_Transform_Anim animations. See if the arms, hands and fingers retarget properly. They are used during paragliding.

- You can find a **Demo_Level_BP** inside the Blueprints folder. It is only used for the Demo. It changes the music volume and camera distance in the Demo.

- **Component Migration:**



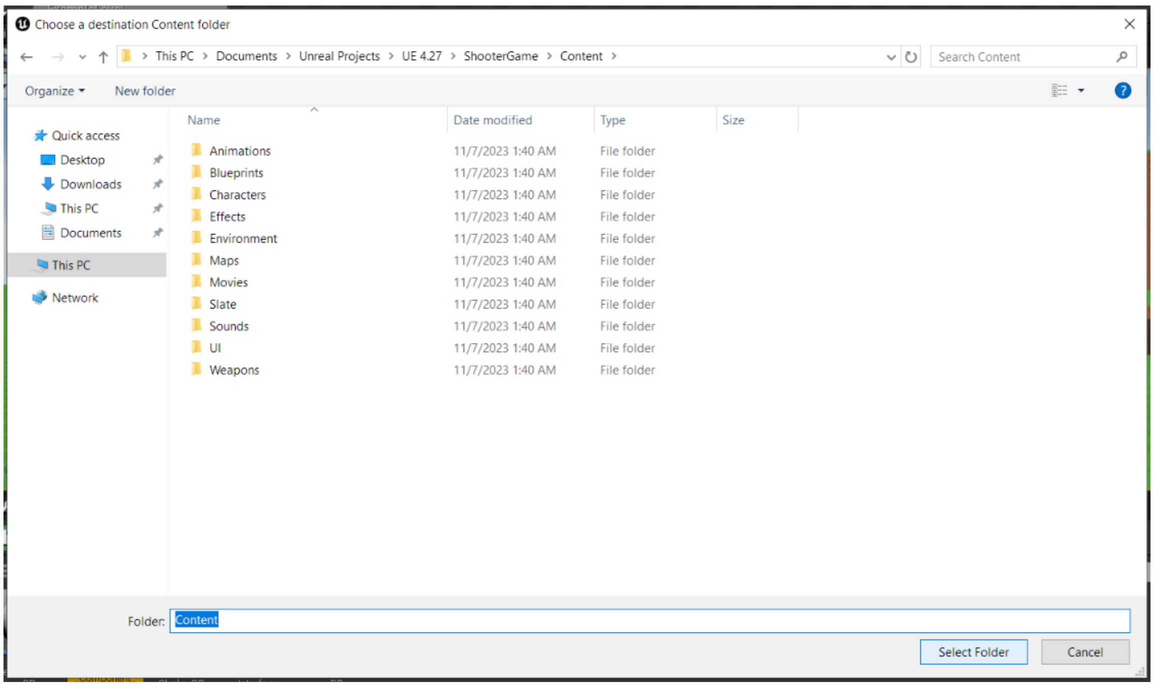
First, open the Skydiving and Paragliding project. Go to the Blueprints folder, then open the Version 2 folder and right click on the Skydiving Component BP. Go to Asset Actions and select Migrate.


Asset Report

The following assets will be migrated to another content folder.

```
        ☑ MI_Hang_Glider_Hardware_2
        ☑ MI_Hang_Glider_Hardware_3
        ☑ MI_Hang_Glider_Wing
        ☑ MI_Hang_Glider_Wing_1
      ☑ Meshes
        ☑ SM_Hang_Glider
      ☑ Textures
        ☑ T_Hang_Glider_D
        ☑ T_Hang_Glider_M
        ☑ T_Tech_Hex_Tile_M
☐ ThirdPersonAssets
  ☑ Mannequin
    ☑ Character
      ☑ Materials
        ☑ M_UE4Man_Body
        ☑ MaterialLayers
          ☑ MF_GlossyBlack_Latex_UE4
          ☑ MF_Plastic_Shiny_Beige
          ☑ MF_Plastic_Shiny_Beige_LOGO
          ☑ MF_SoftMetal_UE4
          ☑ T_ML_Aluminum01
          ☑ T_ML_Aluminum01_N
          ☑ T_ML_Rubber_Blue_01_D
          ☑ T_ML_Rubber_Blue_01_N
        ☑ MI_UE4Man_ChestLogo
      ☑ Mesh
        ☑ SK_Mannequin
```

All its dependencies will be selected. Deselect the **ThirdPersonAssets** folder as we don't need those assets, then click OK.


Choose a destination Content folder

This PC > Documents > Unreal Projects > UE 4.27 > ShooterGame > Content >

Name	Date modified	Type	Size
Animations	11/7/2023 1:40 AM	File folder	
Blueprints	11/7/2023 1:40 AM	File folder	
Characters	11/7/2023 1:40 AM	File folder	
Effects	11/7/2023 1:40 AM	File folder	
Environment	11/7/2023 1:40 AM	File folder	
Maps	11/7/2023 1:40 AM	File folder	
Movies	11/7/2023 1:40 AM	File folder	
Slate	11/7/2023 1:40 AM	File folder	
Sounds	11/7/2023 1:40 AM	File folder	
UI	11/7/2023 1:40 AM	File folder	
Weapons	11/7/2023 1:40 AM	File folder	

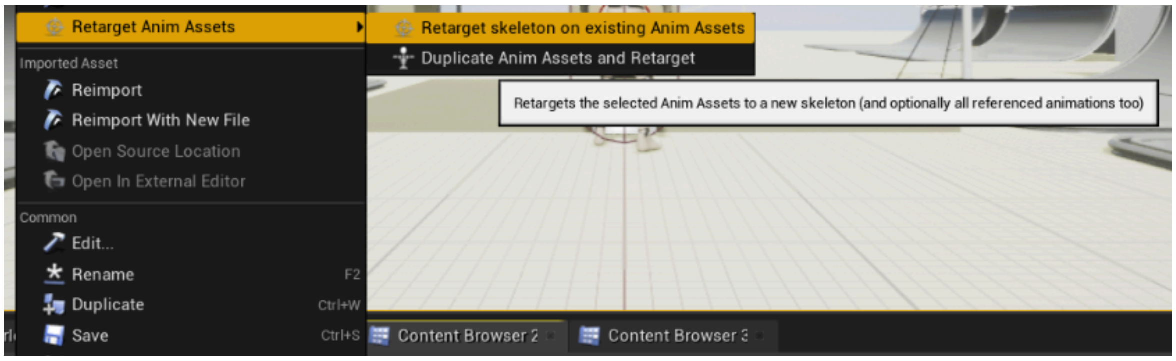
Folder: Content

Select the **Content** folder of your project and click on 'Select Folder'. Now, the component and all its dependencies will be migrated to your project.

- ## Retargeting Animations Inside your Project:

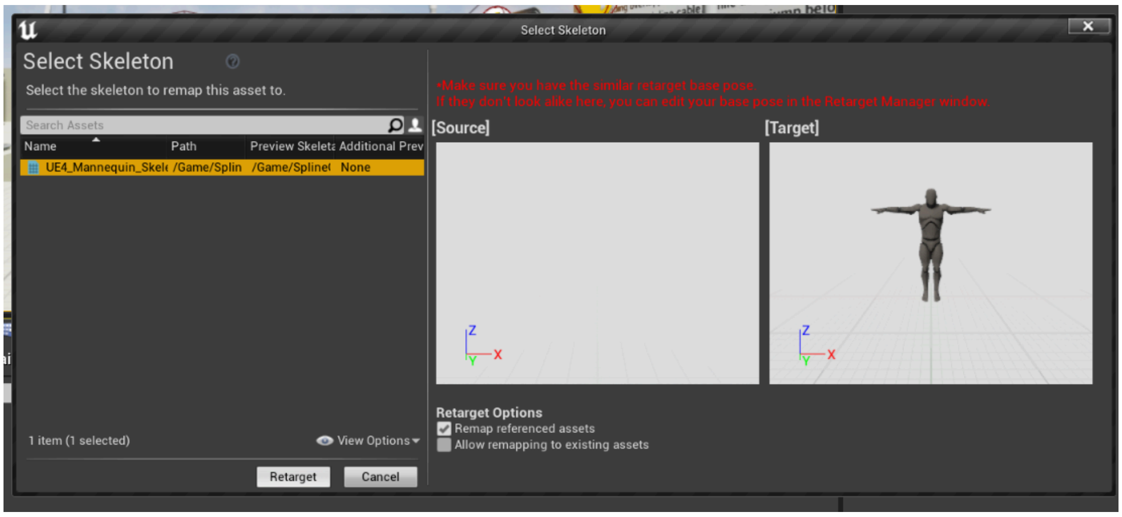
This section is divided into UE 4 / 5. Follow the steps related to your engine version.

Unreal Engine 4:



Retargets the selected Anim Assets to a new skeleton (and optionally all referenced animations too)

Go to the migrated Animations folder. Open the Version 2 folder and select all the animations, montages and Blend Spaces (while holding shift to select them all), then right click on any of them and select **Retarget Anim Assets ⇒ Retarget skeleton on existing Anim Assets.**



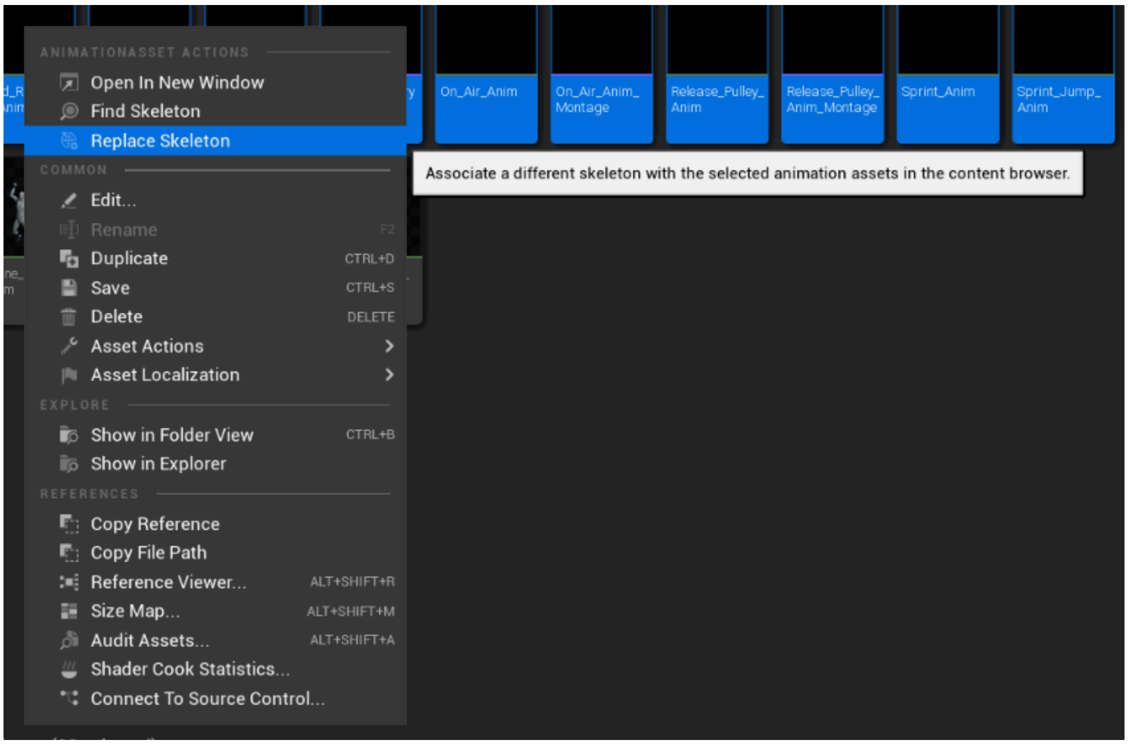
Name	Path	Preview Skeleta	Additional Prev
UE4_Mannequin_Skele	/Game/Splin	/Game/Splinet	None

•Make sure you have the similar retarget base pose.
If they don't look alike here, you can edit your base pose in the Retarget Manager window.

Retarget Options
[✓] Remap referenced assets
[ ] Allow remapping to existing assets

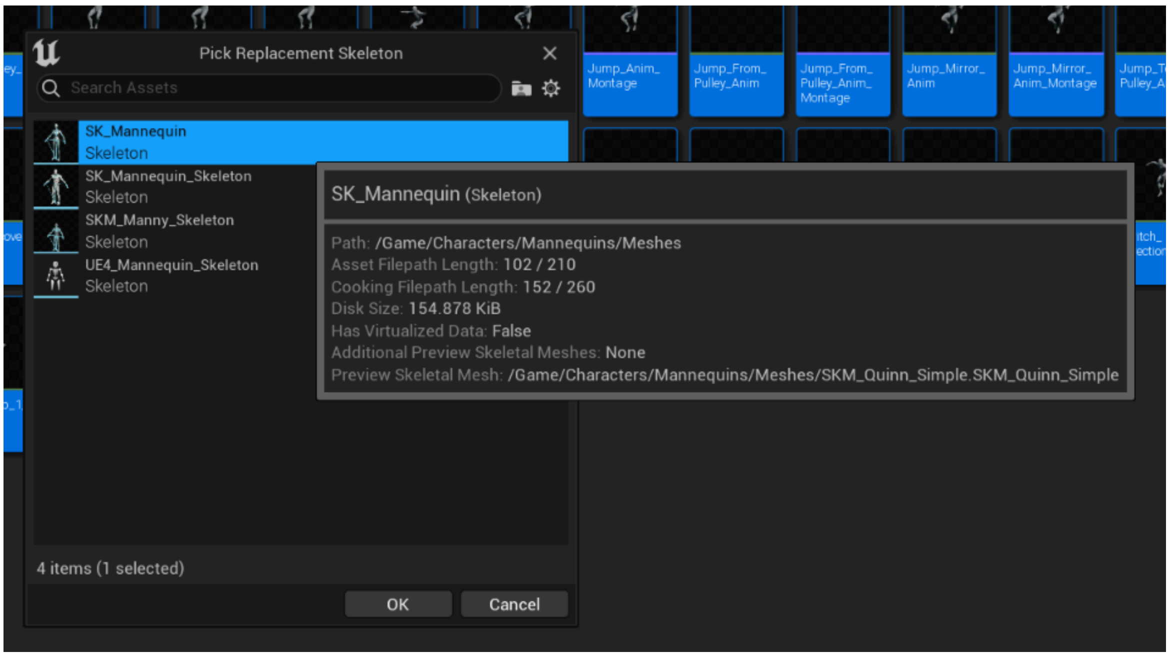
Select your character skeleton and click Retarget. Then **save all the animations, montages and Blend Spaces** before continuing.

Unreal Engine 5:



Associate a different skeleton with the selected animation assets in the content browser.

Go to the migrated Animations folder. Open the Version 2 folder and select all the animations, montages and Blend Spaces (while holding shift to select them all), then right click on any of them and select **Replace Skeleton.**



SK_Mannequin
Skeleton
SK_Mannequin_Skeleton
Skeleton
SKM_Manny_Skeleton
Skeleton
UE4_Mannequin_Skeleton
Skeleton

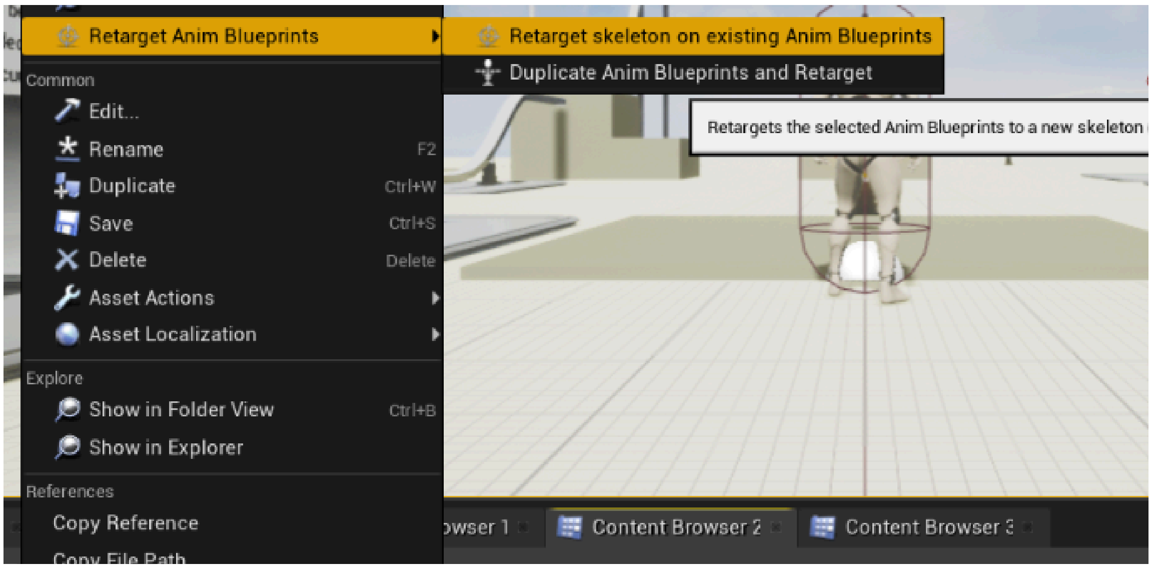
4 items (1 selected)

SK_Mannequin (Skeleton)

Path: /Game/Characters/Mannequins/Meshes
Asset Filepath Length: 102 / 210
Cooking Filepath Length: 152 / 260
Disk Size: 154.878 KiB
Has Virtualized Data: False
Additional Preview Skeletal Meshes: None
Preview Skeletal Mesh: /Game/Characters/Mannequins/Meshes/SKM_Quinn_Simple.SKM_Quinn_Simple

Select your character skeleton and click OK. Then **save all the animations, montages and Blend Spaces** before continuing.

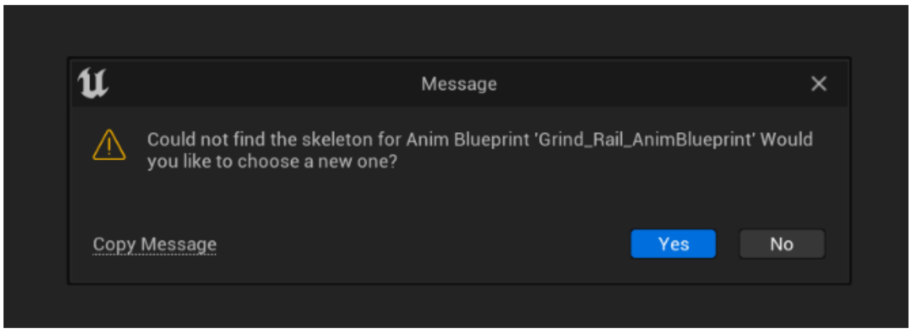
Unreal Engine 4:



Retargets the selected Anim Blueprints to a new skeleton

Open the Animations ⇒ Animation Blueprint folder and right click on the Skydiving Anim Blueprint, then select **Retarget Anim Blueprints** ⇒ **Retarget skeleton on existing Anim Blueprints** and select the character skeleton. Click Retarget. Now open the animation blueprint (there should be no compile errors if the previous animations and Blend Spaces were retargeted and saved). You can save and close the blueprint.

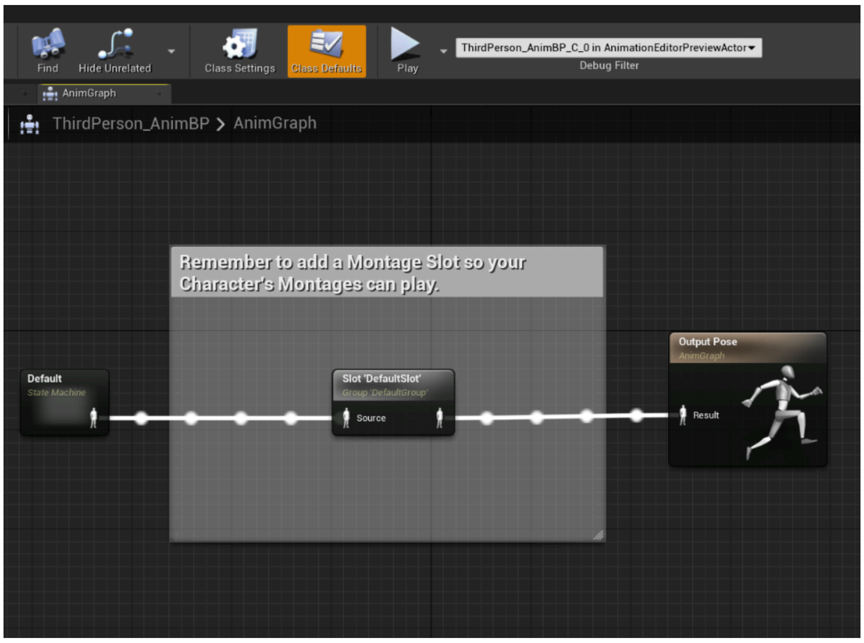
Unreal Engine 5:



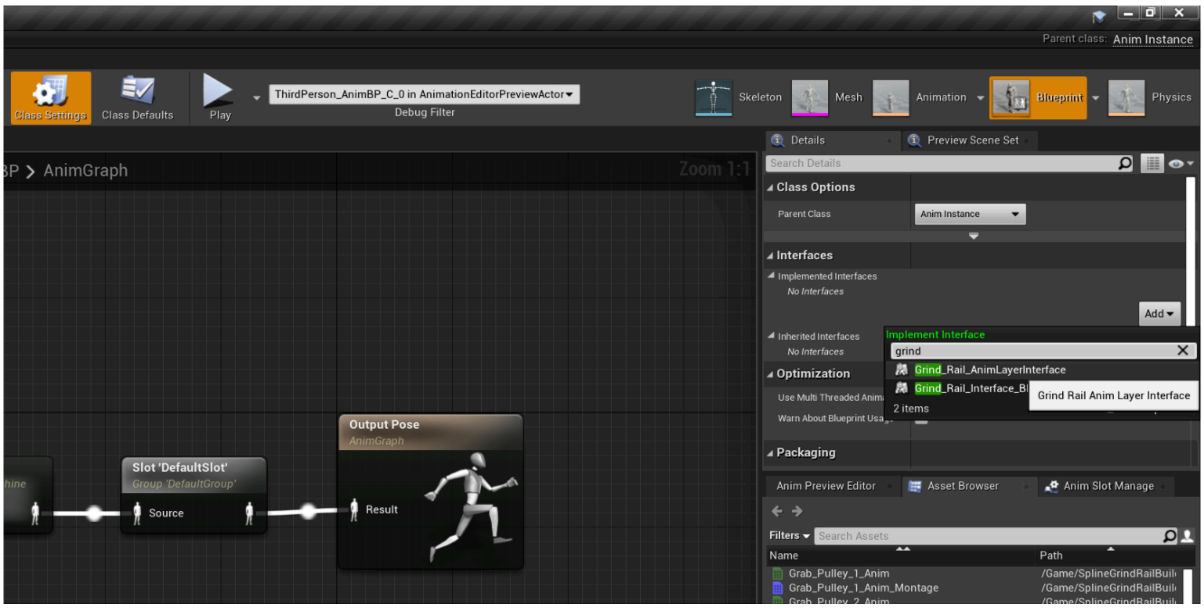
Could not find the skeleton for Anim Blueprint 'Grind_Rail_AnimBlueprint' Would you like to choose a new one?

Open the Animations ⇒ Animation Blueprint folder and double click on the Skydiving Anim Blueprint. A message will ask if you would like to choose a new skeleton. Click Yes and select the character skeleton, then click OK. Now open the animation blueprint (there should be no compile errors if the previous animations and Blend Spaces were retargeted and saved). You can save and close the blueprint.

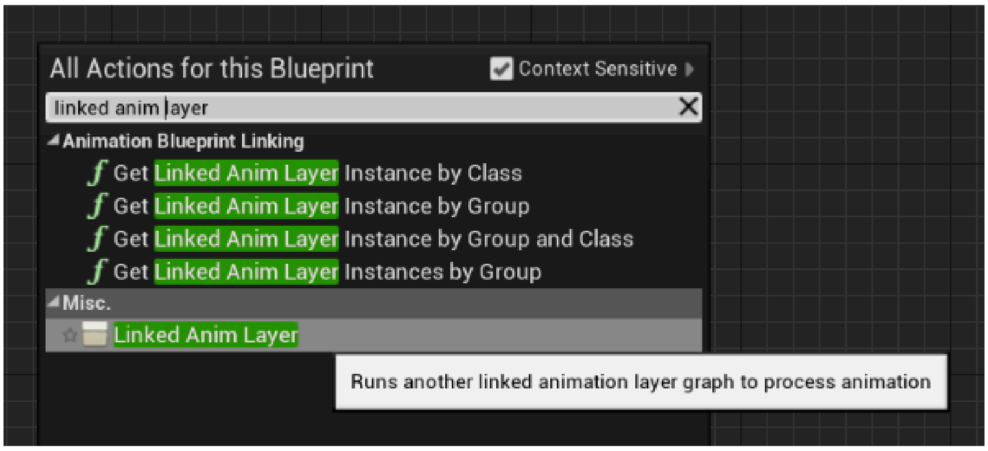
- **Main Animation Blueprint:**



Open your Character's Animation Blueprint. Make sure to have a **DefaultSlot** node as it is used to play the Montages. If your Animation BP doesn't have one, it can be created by right clicking and typing in - Slot 'DefaultSlot' - and selecting that option. Connect it like in the screenshot above.



Select *Class Settings* in the top bar menu and, in the *Interfaces* settings (to the right), click on Add and select the **Skydiving Anim Layer Interface**. Then **compile the blueprint**.

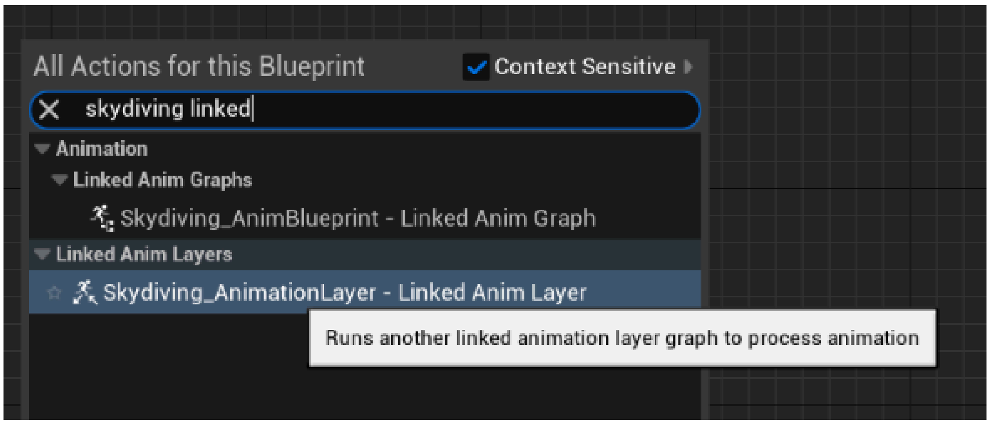

All Actions for this Blueprint ☑ Context Sensitive

linked anim layer

Animation Blueprint Linking
- Get Linked Anim Layer Instance by Class
- Get Linked Anim Layer Instance by Group
- Get Linked Anim Layer Instance by Group and Class
- Get Linked Anim Layer Instances by Group

Misc.
- Linked Anim Layer

Runs another linked animation layer graph to process animation


All Actions for this Blueprint ☑ Context Sensitive

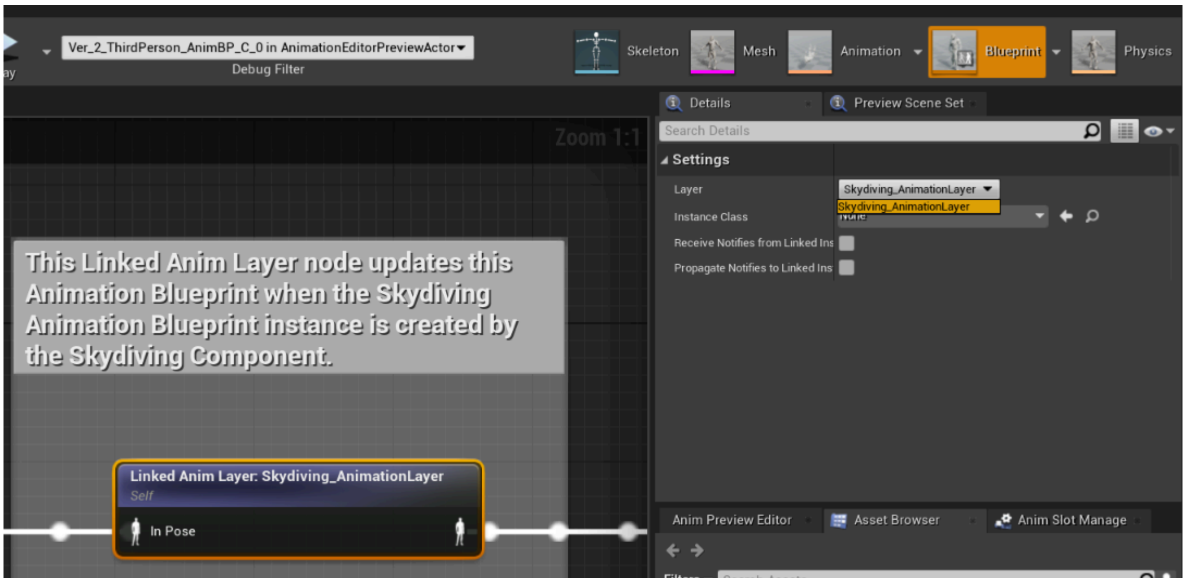
skydiving linked

Animation
- Linked Anim Graphs
  - Skydiving_AnimBlueprint - Linked Anim Graph

Linked Anim Layers
- Skydiving_AnimationLayer - Linked Anim Layer

Runs another linked animation layer graph to process animation

Right click anywhere on the graph and type *Linked Anim Layer* (in Unreal Engine 5 type *Skydiving Linked*). Select the highlighted option like in the screenshots above (the Anim Layer, not the Anim Graph).



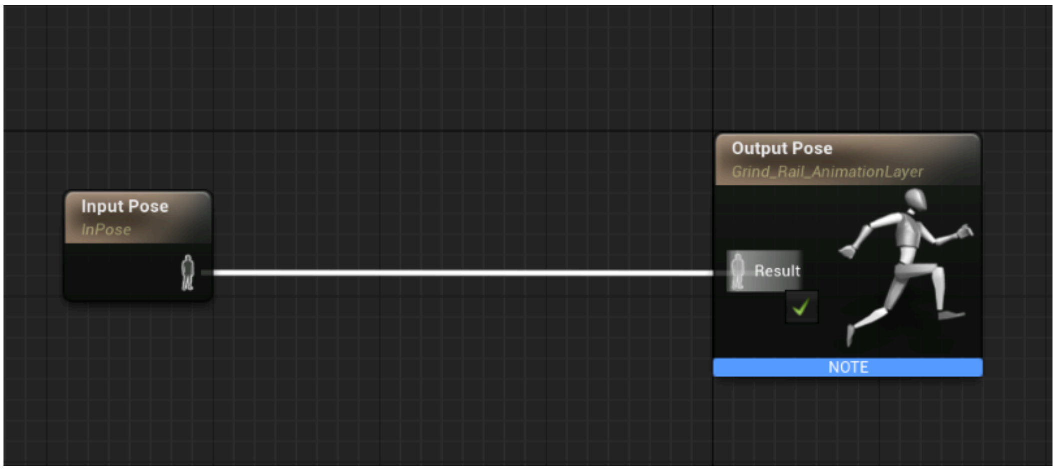
This Linked Anim Layer node updates this Animation Blueprint when the Skydiving Animation Blueprint instance is created by the Skydiving Component.

Linked Anim Layer: Skydiving_AnimationLayer
Self
In Pose

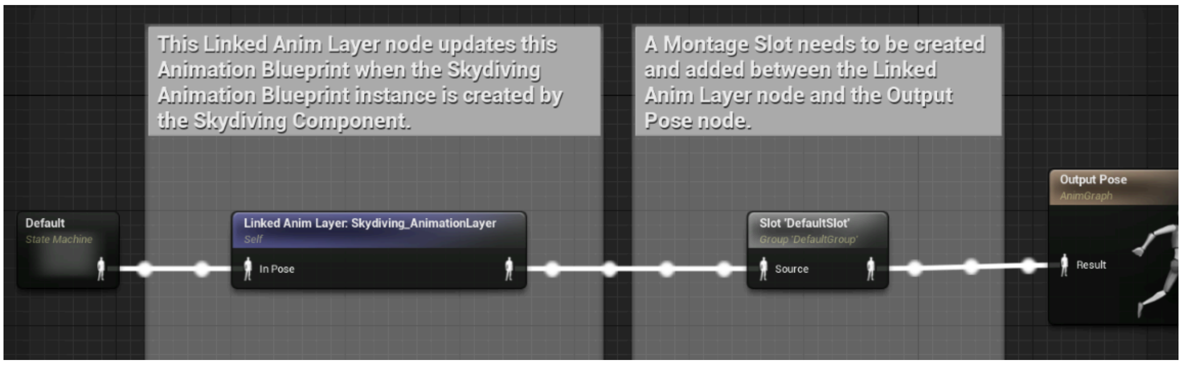
Settings

Layer	Skydiving_AnimationLayer
Instance Class	None
Receive Notifies from Linked Ins	☐
Propagate Notifies to Linked Ins	☐

*** In Unreal Engine 4 only**, select the created node and, in the ***Layer*** (not Instance Class) drop down menu to the right, select Skydiving AnimationLayer. The node will be updated automatically.



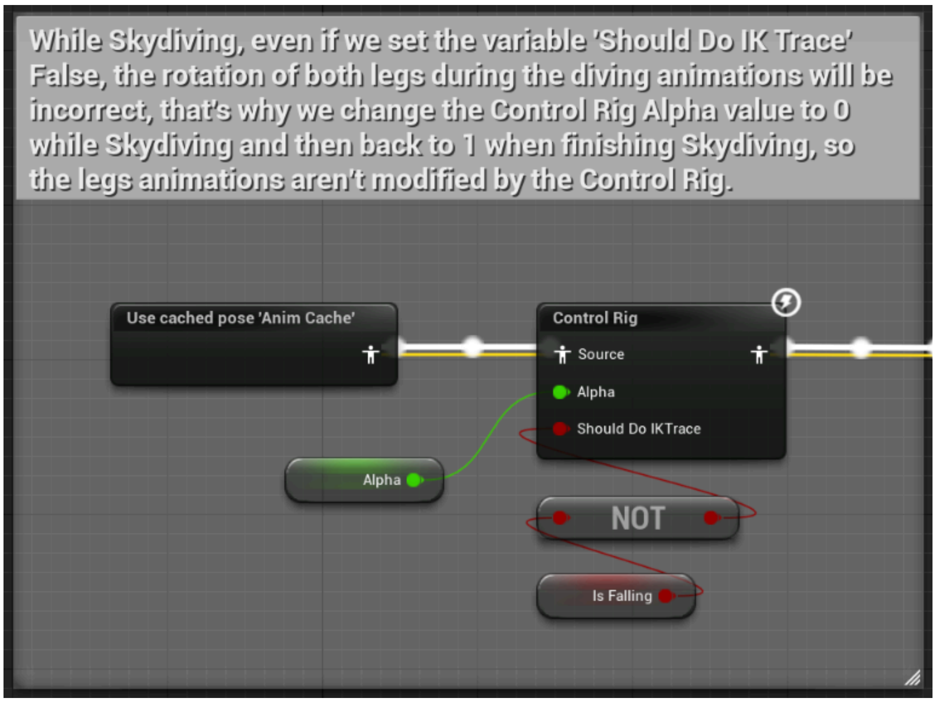
Double click on the new Linked Anim Layer node to open it. Then connect the Input Pose with the Output Pose.


This Linked Anim Layer node updates this Animation Blueprint when the Skydiving Animation Blueprint instance is created by the Skydiving Component.

A Montage Slot needs to be created and added between the Linked Anim Layer node and the Output Pose node.

Return to the main Animation Graph and connect the Linked Anim Layer node to the State Machine (or its cache) and to the Montage Default Slot, so it updates **in between them**. Compile the blueprint one more time and save.

- - - - Follow this part only if you are using Unreal Engine’s Control Rig Foot IK solution - - - -

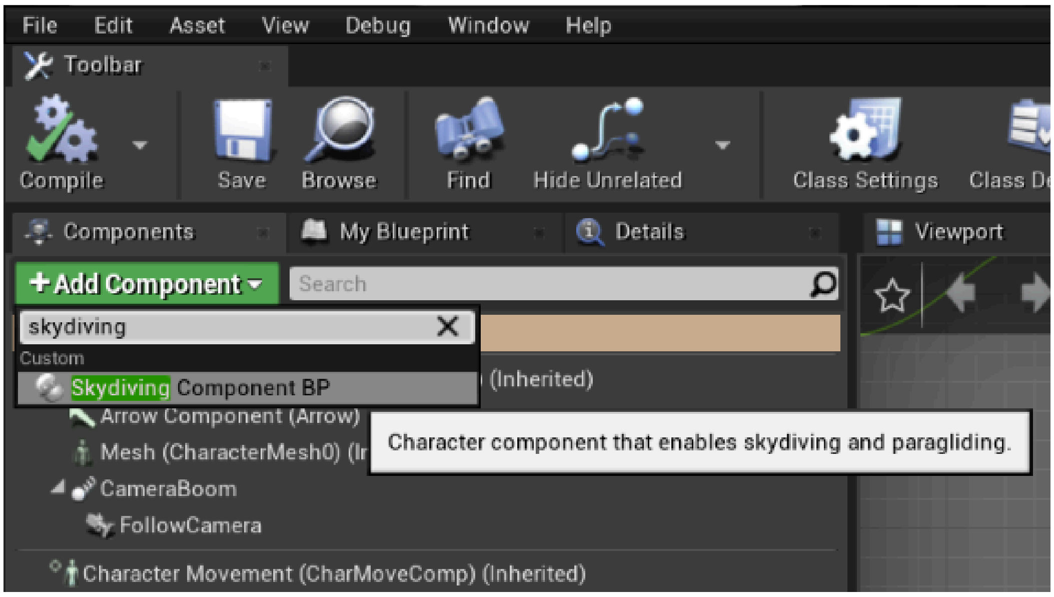

While Skydiving, even if we set the variable 'Should Do IK Trace' False, the rotation of both legs during the diving animations will be incorrect, that's why we change the Control Rig Alpha value to 0 while Skydiving and then back to 1 when finishing Skydiving, so the legs animations aren't modified by the Control Rig.

While using a foot IK solution it is possible that both legs rotate incorrectly while skydiving (even if the variable -Is Falling- is False). To fix this we need to change the Alpha value used by the Control Rig. To do so, we expose it as a variable so it can be used inside your Character's BP. There, we set the value to 0 while paragliding / skydiving and return it to 1 when finishing.

In the Control Rig node, right click on ‘Alpha’ and select -Promote to Variable-. Then compile the blueprint and save.

- -

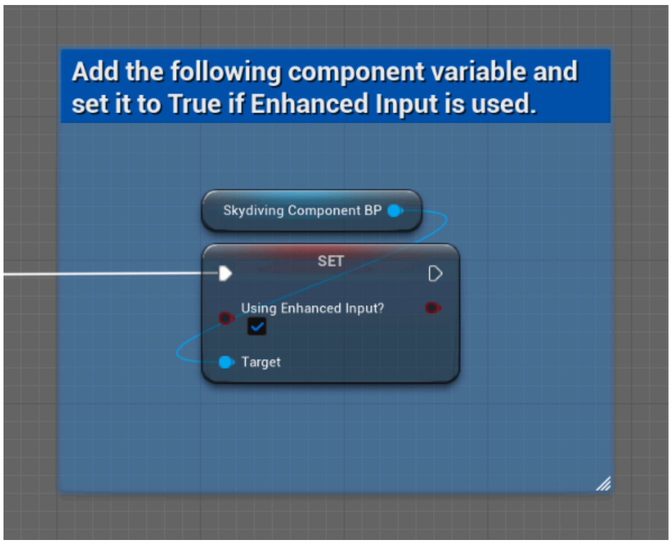
- **Inside your Character's Blueprint:**

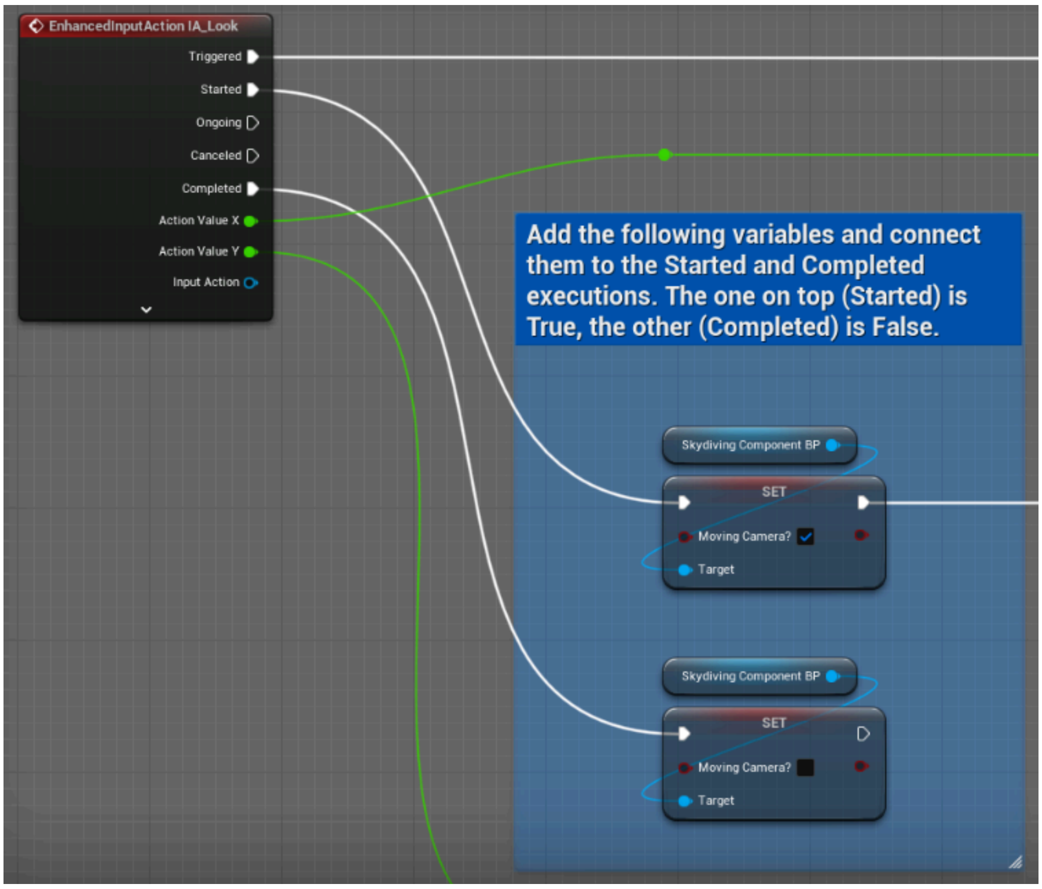


Character component that enables skydiving and paragliding.

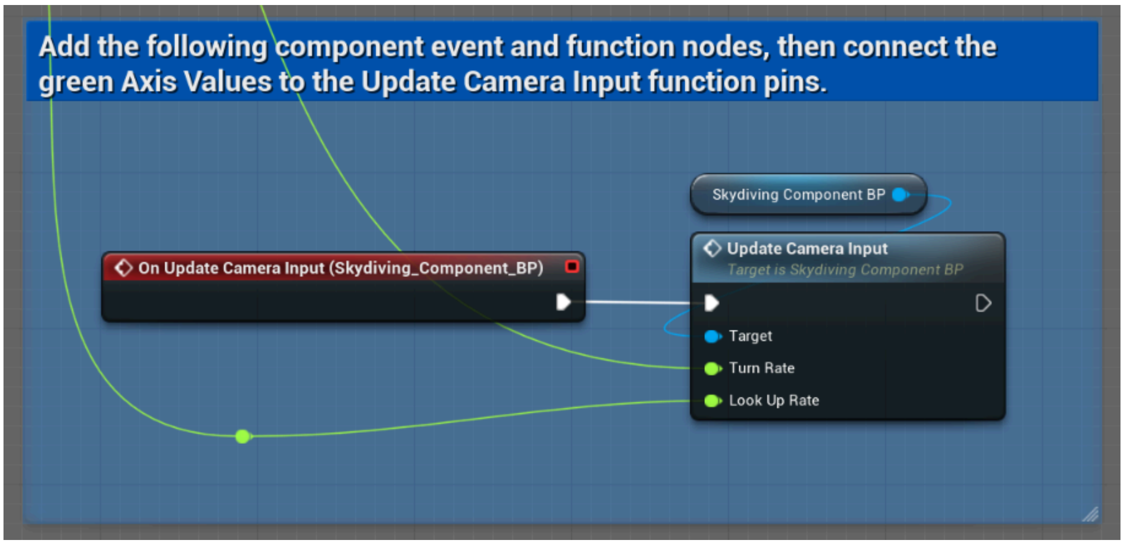
Open your Character's BP, click on +Add Component and select the Skydiving Component BP. **Avoid changing its name for now.** Then compile the blueprint.

From now you can open the **Ver 2 Third Person Character BP** that's included in the example project (Demo ⇒ ThirdPersonAssets ⇒ Blueprints), so you can copy the nodes and paste them directly into your Character's Blueprint. This will work as long as the component is named the same in between blueprints.

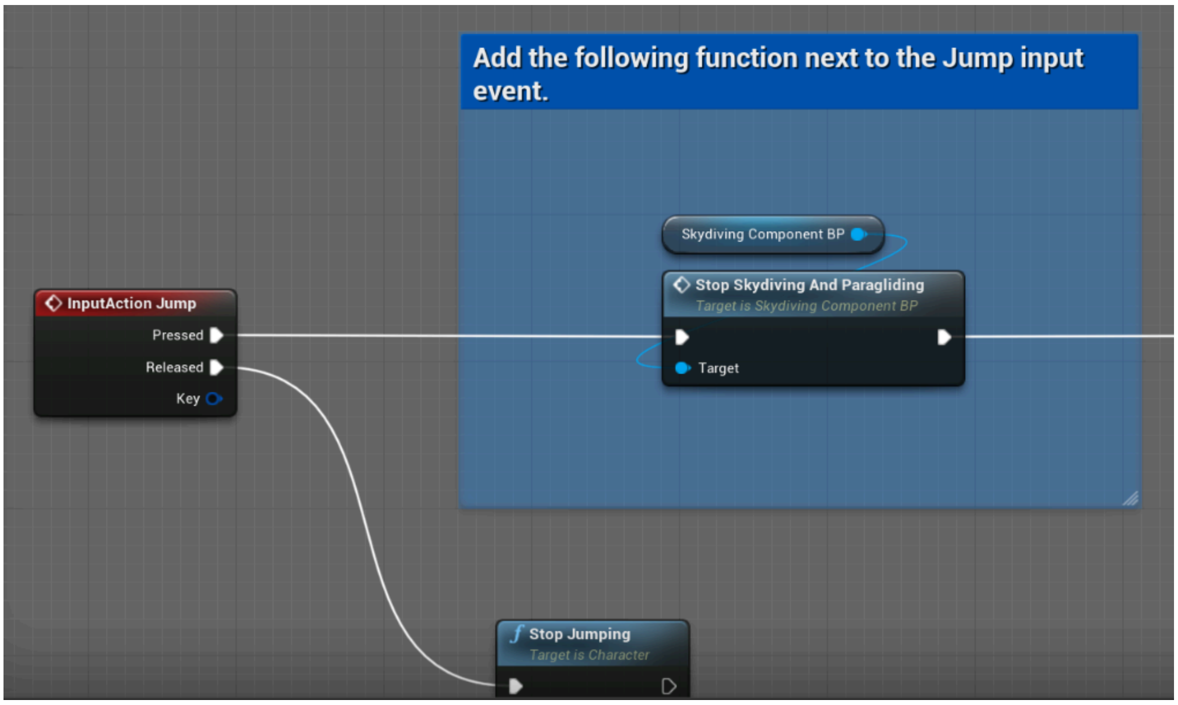

Add the following component variable and set it to True if Enhanced Input is used.


Add the following variables and connect them to the Started and Completed executions. The one on top (Started) is True, the other (Completed) is False.

If you **are using Enhanced Input** to move the Camera, go to Begin Play and copy the boolean -Using Enhanced Input?- connect it and set it to True. Then go to the Camera Input (Look) section and copy the booleans -Moving Camera?- and connect them to the Started and Completed executions. Set them to True and False as shown in the screenshot above.


Add the following component event and function nodes, then connect the green Axis Values to the Update Camera Input function pins.
Skydiving Component BP
Update Camera Input
Target is Skydiving Component BP
On Update Camera Input (Skydiving_Component_BP)
Target
Turn Rate
Look Up Rate

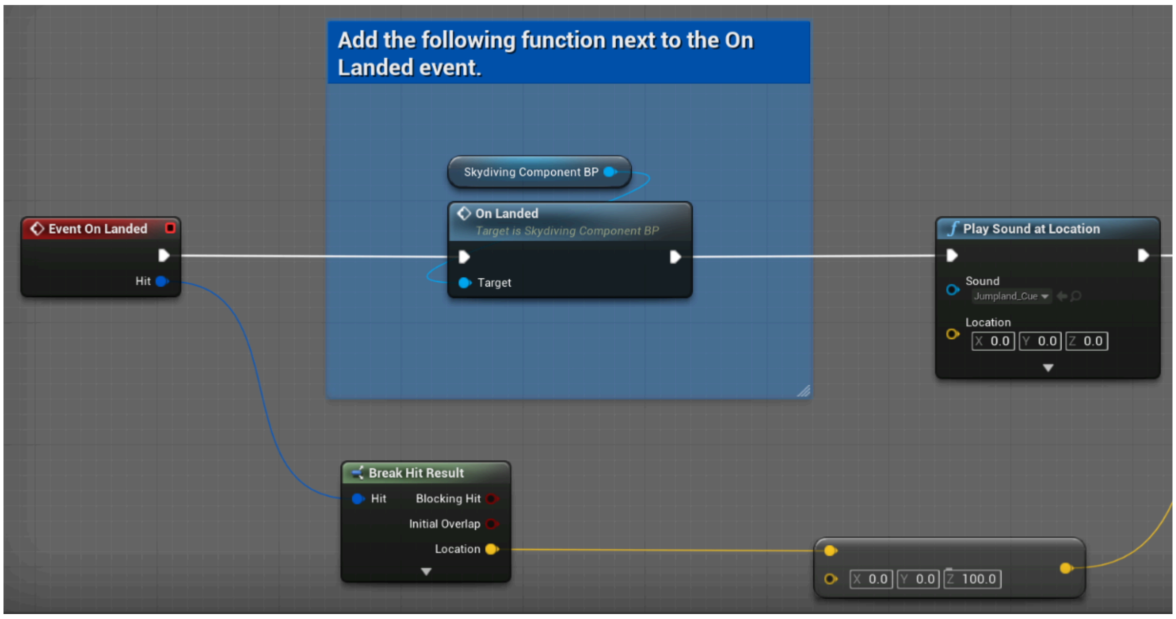
If you **are not using Enhanced Input**, copy - paste the event and function shown in the screenshot above and connect the green Axis Values (Input Axis Turn Rate and Look Up Rate).


Add the following function next to the Jump input event.
Skydiving Component BP
Stop Skydiving And Paragliding
Target is Skydiving Component BP
Target
InputAction Jump
Pressed
Released
Key
Stop Jumping
Target is Character

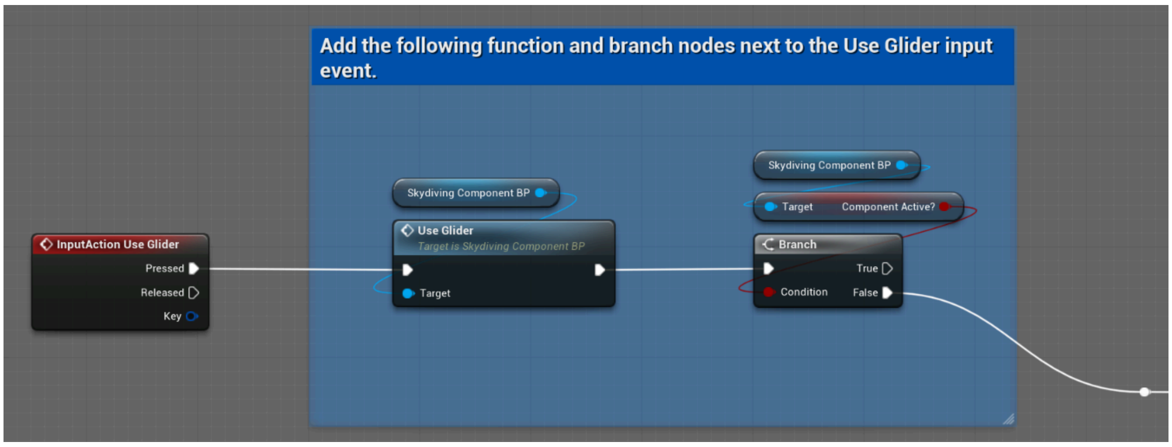
Now go to your Character's Jump Input Event and add the function node shown in the screenshot above. Make sure that it is connected to the 'Pressed' ('Started' in Enhanced Input) execution and placed **before the Jump Function**.

Depending on the logic already present in your character blueprint, there can be more nodes connected in between the Input and the -**Stop Skydiving And Paragliding**- function. This function will stop both systems if they are active mid air, otherwise it won't trigger any logic.

You can also connect the function to other Input events. It works while falling only.

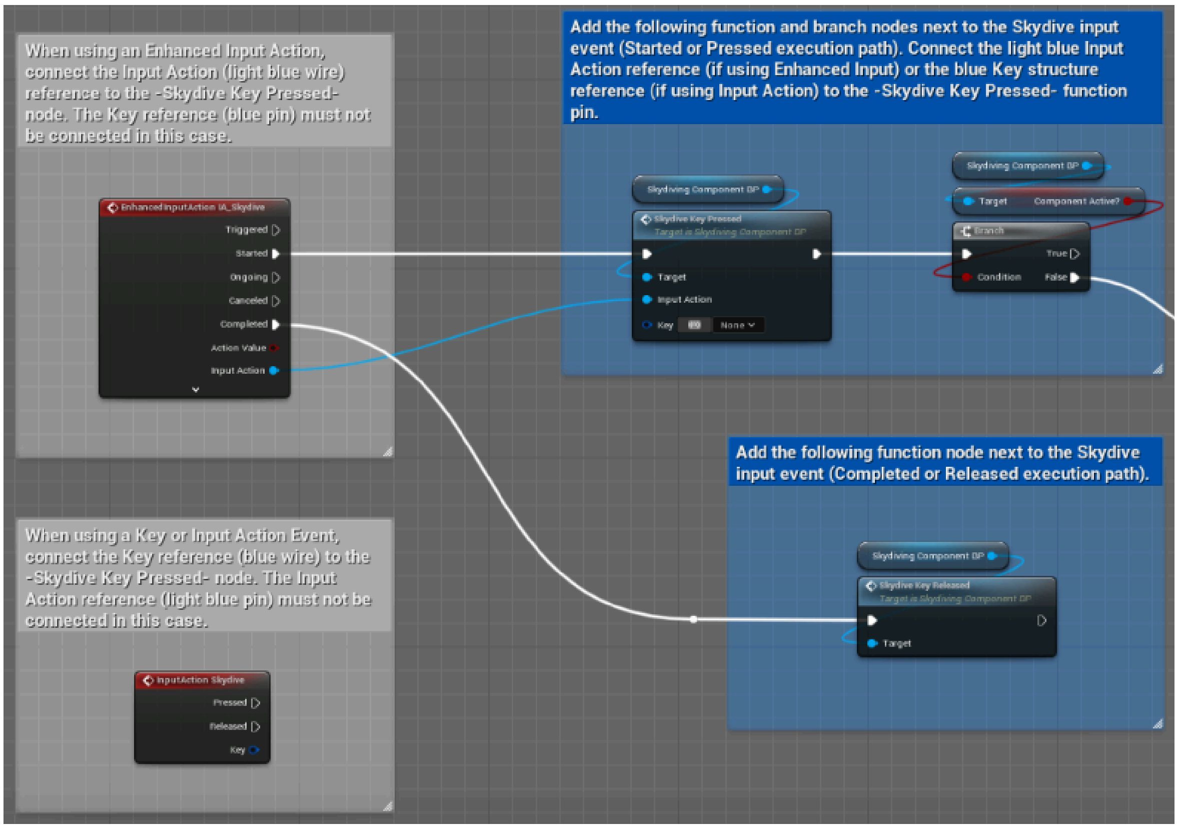


The Event On Landed also stops both systems if they are active. Here we must add the **-On Landed-** component function.



Now go to the Input Event that will be assigned to the function **-Use Glider-** and connect it to the Pressed (Started) execution.

Important: Here we will use the branch with the boolean **-Component Active?-** so any logic that follows the False path will only trigger when the Skydiving Component is not active (no skydiving or paragliding active). This is useful in case we need to use the same input for other actions or logic and we don't want them to trigger while skydiving or paragliding.


When using an Enhanced Input Action, connect the Input Action (light blue wire) reference to the -Skydive Key Pressed- node. The Key reference (blue pin) must not be connected in this case.

Add the following function and branch nodes next to the Skydive input event (Started or Pressed execution path). Connect the light blue Input Action reference (if using Enhanced Input) or the blue Key structure reference (if using Input Action) to the -Skydive Key Pressed- function pin.

Add the following function node next to the Skydive input event (Completed or Released execution path).

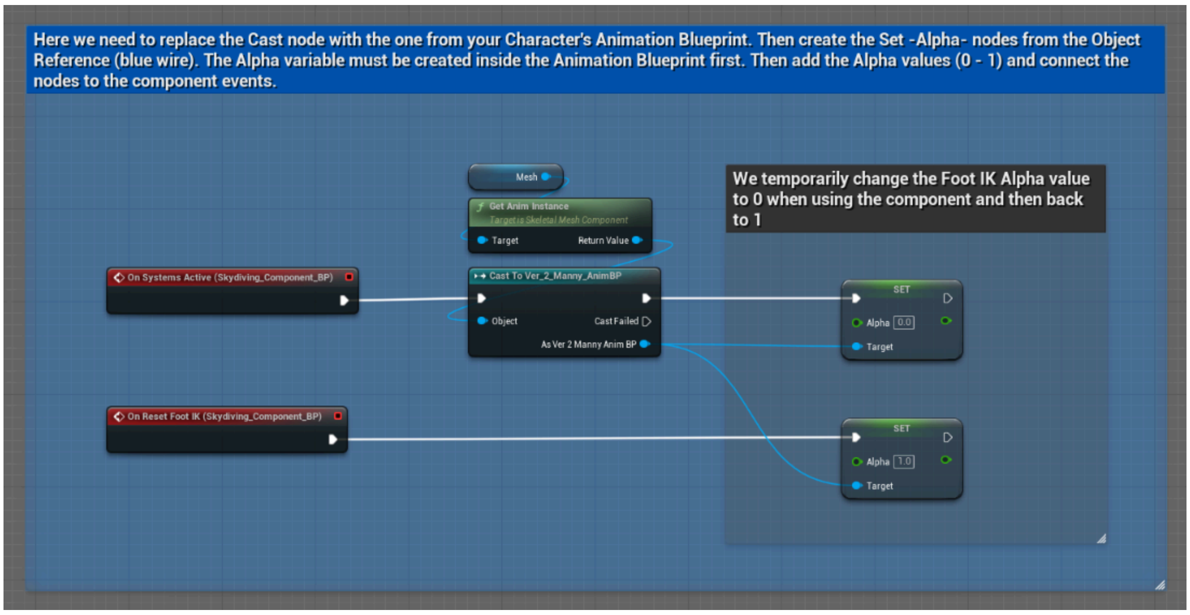
When using a Key or Input Action Event, connect the Key reference (blue wire) to the -Skydive Key Pressed- node. The Input Action reference (light blue pin) must not be connected in this case.

Look for an Input Event that will be assigned to the function **-Skydive Key Pressed-** and connect it to the Pressed (Started) execution. This will begin the Skydiving action. Also add the branch with the boolean, like in the previous gliding section.

If using Enhanced Input, connect the **light blue Input Action** reference to the function. If not using Enhanced Input, just connect the **blue Key Structure** reference.

Connect the **-Skydive Key Released-** to the Released (Completed) execution.

- - - - Follow this part only if you are using Unreal Engine’s Control Rig Foot IK solution - - - -


Here we need to replace the Cast node with the one from your Character's Animation Blueprint. Then create the Set -Alpha- nodes from the Object Reference (blue wire). The Alpha variable must be created inside the Animation Blueprint first. Then add the Alpha values (0 - 1) and connect the nodes to the component events.

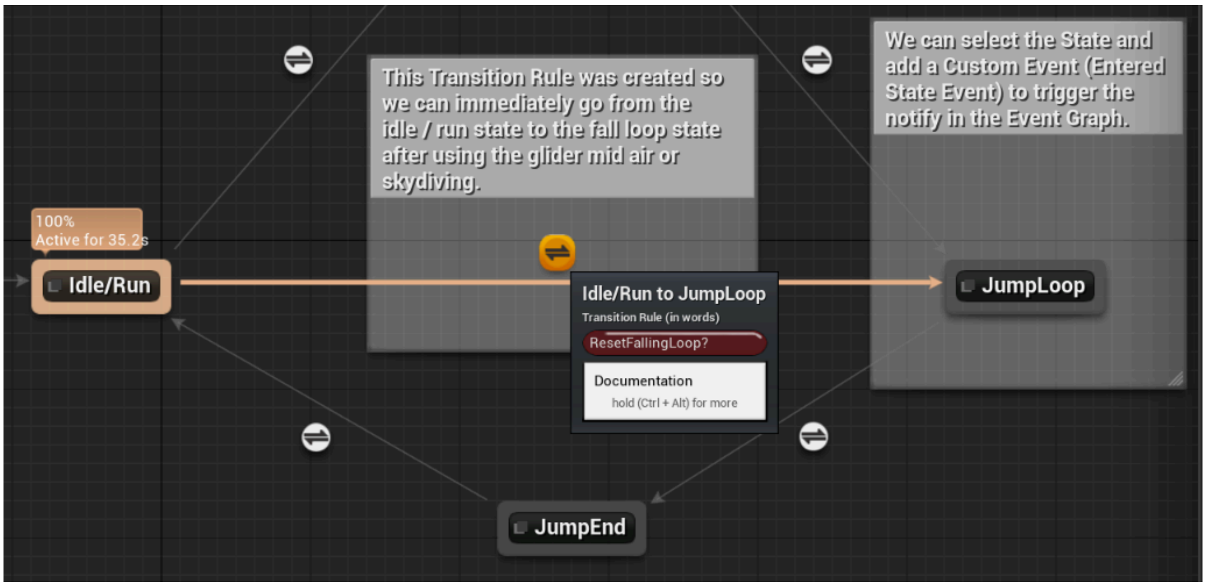
We temporarily change the Foot IK Alpha value to 0 when using the component and then back to 1

Mesh
Get Anim Instance
Target is Skeletal Mesh Component
Target
Return Value
On Systems Active (Skydiving_Component_BP)
Cast To Ver_2_Manny_AnimBP
Object
Cast Failed
As Ver 2 Manny Anim BP
SET
Alpha 0.0
Target
On Reset Foot IK (Skydiving_Component_BP)
SET
Alpha 1.0
Target

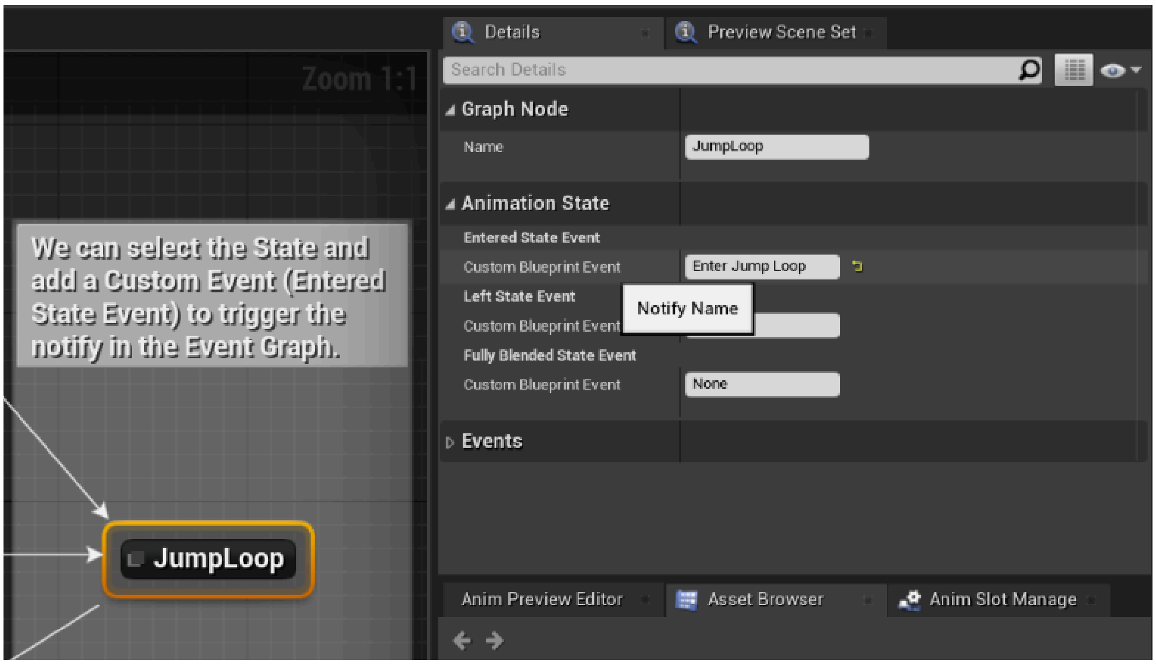
Here we are going to use the previously created Alpha variable. You can copy / paste both events but **the Cast node cannot be pasted from another blueprint**. To create them, first add a reference of your Character´s Skeletal Mesh and from it create a -Get Anim Instance- node and from that one create a -Cast To- node. The -Cast To- node should be casting your Character´s Animation Blueprint. From the **light blue** colored reference create two -Set Alpha- nodes and set their values to 0 and 1. Then connect them as shown in the screenshot above (only the one with 0 is connected to the cast node).

- -

Important: If we notice that, after using the paraglider or ending skydiving mid air, the Character doesn't use the Fall Loop (or Jump Loop) animation State immediately (and instead plays another State, like Jumping), we can create a boolean to help transition immediately to the Fall Loop State. This usually doesn't happen with the new UE 5 State Machine, only with the UE 4 one.

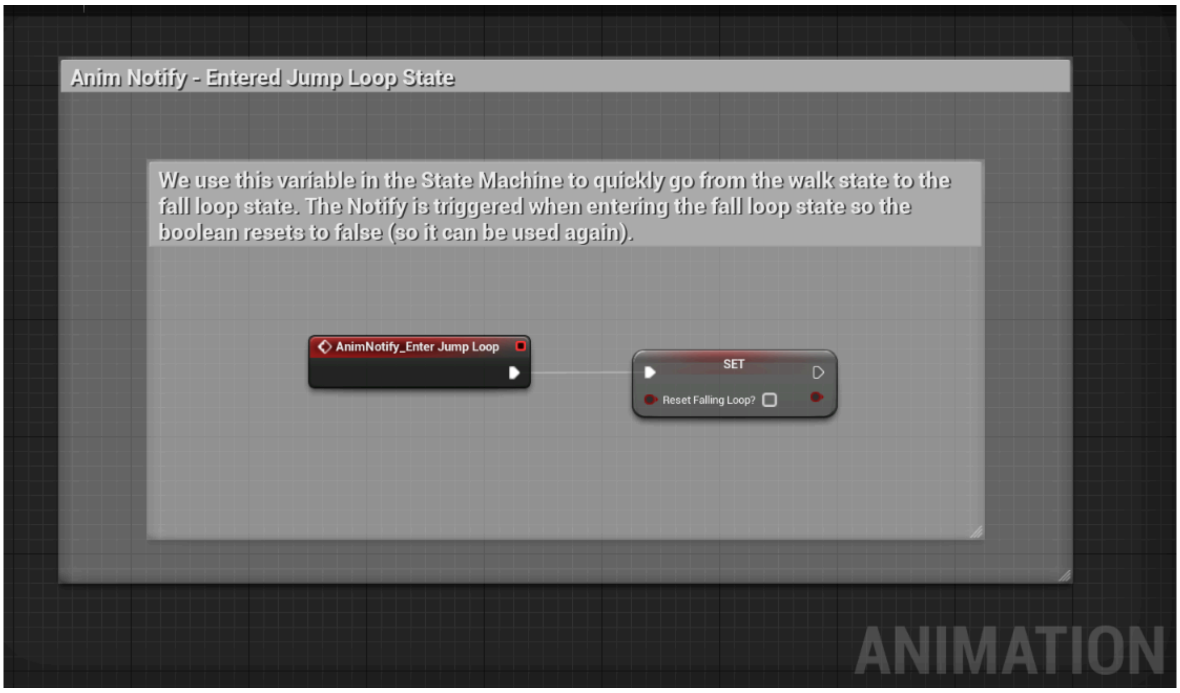

This Transition Rule was created so we can immediately go from the idle / run state to the fall loop state after using the glider mid air or skydiving.
We can select the State and add a Custom Event (Entered State Event) to trigger the notify in the Event Graph.
100% Active for 35.2s
Idle/Run
Idle/Run to JumpLoop
Transition Rule (in words)
ResetFallingLoop?
Documentation
hold (Ctrl + Alt) for more
JumpLoop
JumpEnd

We can create a boolean set to False by default. In the example it is named -**Reset Falling Loop?**-. We then create a transition rule between the Idle / Run (Walking) State to the Fall Loop (Jump Loop) State. We open it and connect the created boolean to the -Can Enter Transition- pin (Result node).


Zoom 1:1
We can select the State and add a Custom Event (Entered State Event) to trigger the notify in the Event Graph.
JumpLoop
Details
Preview Scene Set
Search Details
Events
Anim Preview Editor
Asset Browser
Anim Slot Manage

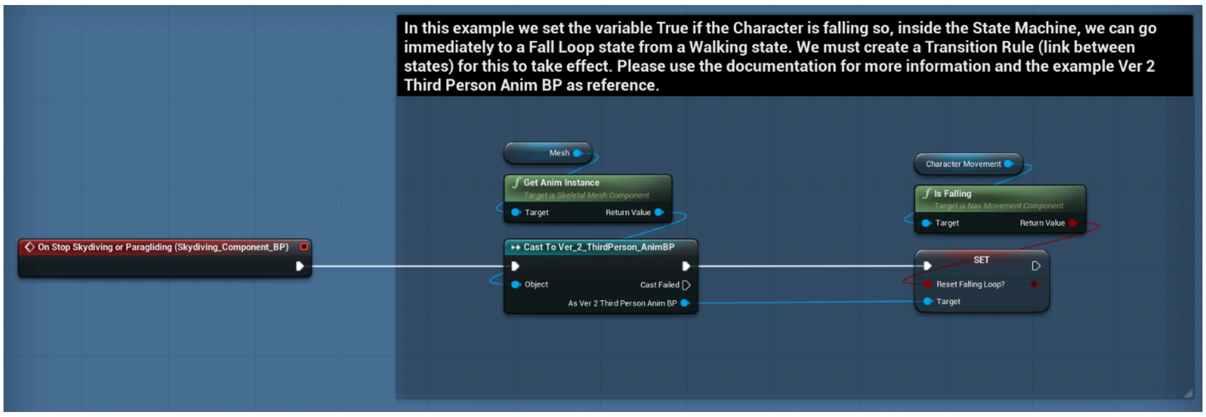
Graph Node
Name: JumpLoop
Animation State
Entered State Event
Custom Blueprint Event: Enter Jump Loop
Left State Event
Custom Blueprint Event
Notify Name
Fully Blended State Event
Custom Blueprint Event: None

Select the Fall Loop (Jump Loop) State and type a name for the **Entered State Event** (to the right), like in the screenshot above. It will trigger when entering the selected State.


Anim Notify - Entered Jump Loop State

We use this variable in the State Machine to quickly go from the walk state to the fall loop state. The Notify is triggered when entering the fall loop state so the boolean resets to false (so it can be used again).

Compile the animation blueprint and go to the Event Graph. Right click anywhere in the graph and type the Name you assigned previously to the Entered State Event and add the event. Connect the created boolean and set it to False. Compile the blueprint and save.

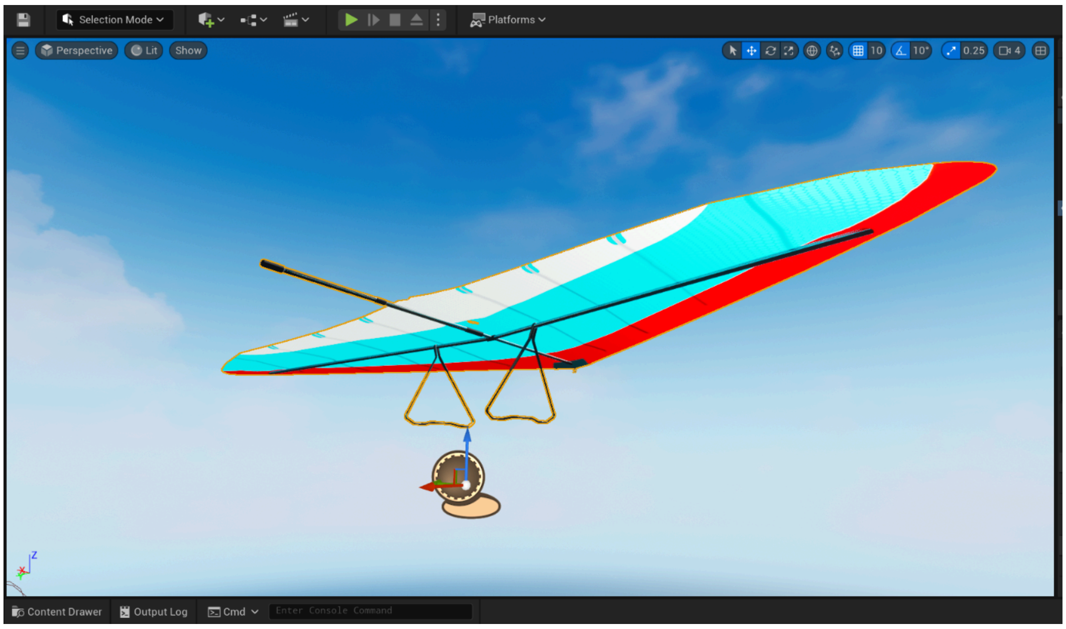

In this example we set the variable True if the Character is falling so, inside the State Machine, we can go immediately to a Fall Loop state from a Walking state. We must create a Transition Rule (link between states) for this to take effect. Please use the documentation for more information and the example Ver 2 Third Person Anim BP as reference.

On Stop Skydiving or Paragliding (Skydiving_Component_BP)
Mesh
Get Anim Instance
Target is Skeletal Mesh Component
Target
Return Value
Cast To Ver_2_ThirdPerson_AnimBP
Object
Cast Failed
As Ver 2 Third Person Anim BP
Character Movement
Is Falling
Target is Nav Movement Component
Target
Return Value
SET
Reset Falling Loop?
Target

Return to the Character blueprint and copy the component event **-On Stop Skydiving or Paragliding-** and connect to it the previously created boolean (from the animation blueprint). To the boolean connect the -Is Falling- character movement function, like in the screenshot above.

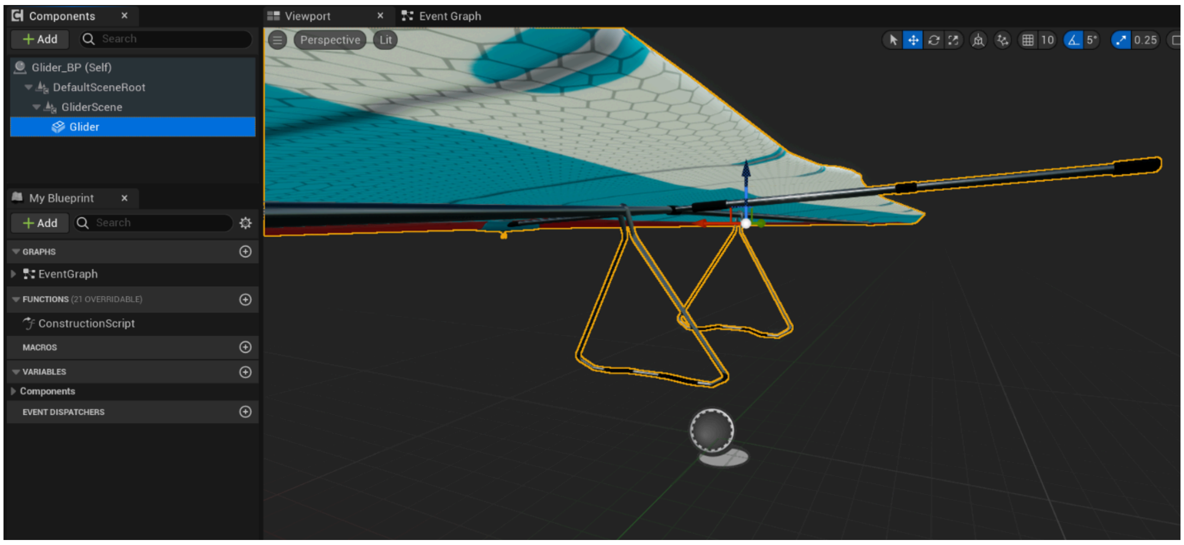
By doing this, everytime the Character stops paragliding or skydiving mid air, the boolean will be set to True and the Fall Loop State will be used immediately (skipping other States in between). The created Entered State Event (Notify) will trigger and reset the value to False so it can be used again later.

- **In the Level Viewport:**



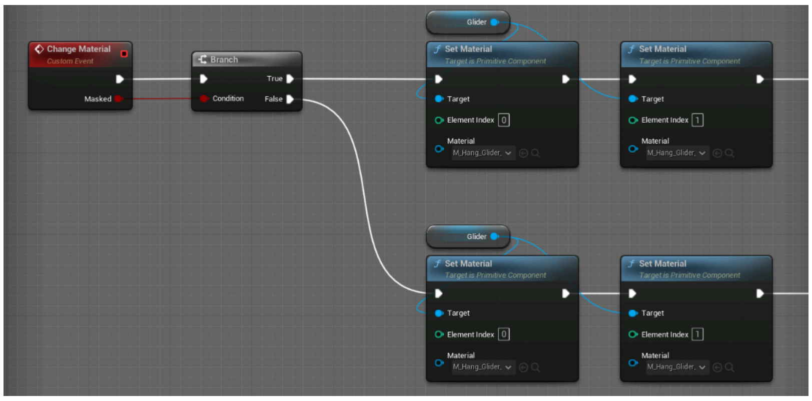
Now, go to your Level's World and **add a Ver 2 Glider BP in the same level the Character is** placed so the Component can save its reference during Begin Play and use it. <u>The Component won't work without it.</u> The Glider is found in the Blueprints ⇒ Version 2 folder.

- **Using Other Glider Meshes:**

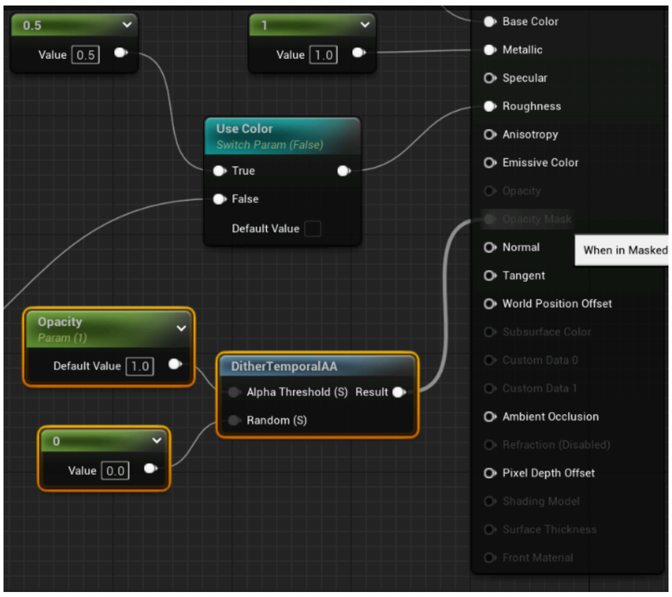


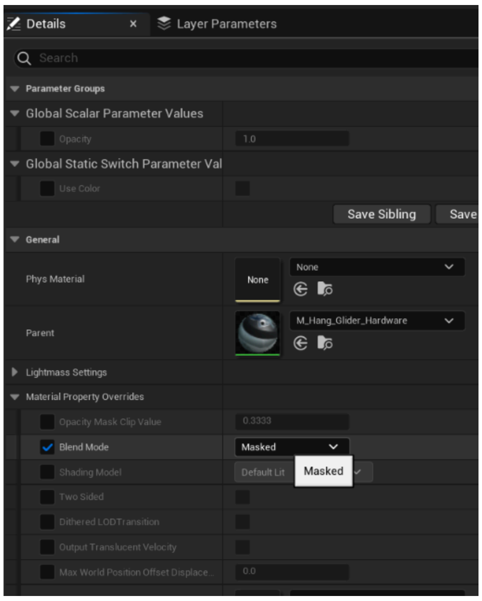
When using other glider meshes, change the mesh from the Glider Static Mesh component. Make sure to adjust the rotation so it faces the same direction the example Glider faces.

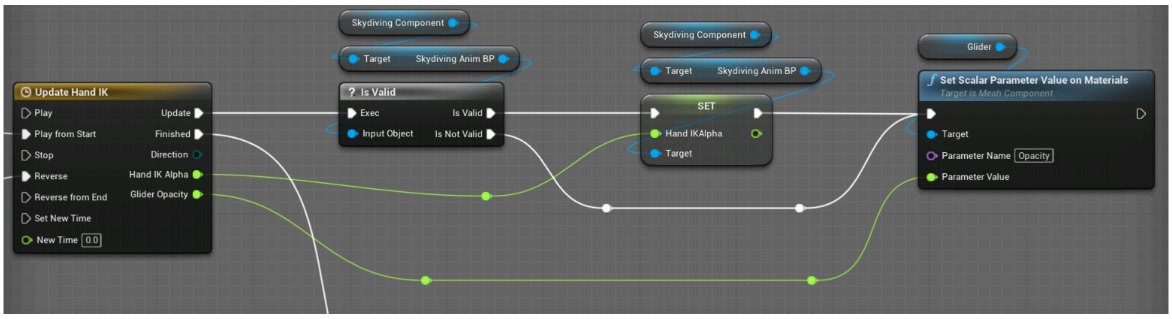
After changing the model, go to the Event Graph and look for the nodes that spawn the Air Trails Niagara System. We need to adjust the Air Trails spawn locations for the new model.



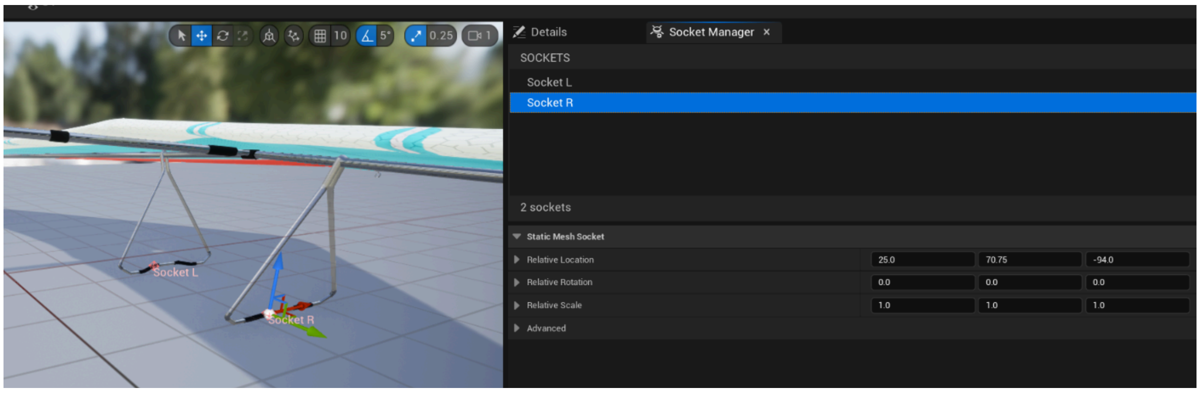
The Event -Change Material- is triggered by the Component. It switches the Glider Mesh Materials from Masked to Opaque. When using other models, change here the material references with the ones used by your model. The nodes connected to the True branch use **masked** instance references. The False branch uses **opaque** instances.







Inside your Glider's Material, add the nodes shown in the picture above (Dither Temporal AA and a float parameter, named Opacity in the example) and create a masked instance of that Material so it can be used with the -Change Material- event. The Opacity value is changed during the *Update Hand IK* Timeline of the Glider blueprint. The name used by the float parameter inside the Material (in the example it's named Opacity) must be the same one typed in the **-Set Scalar Parameter Value on Materials-** function.


SOCKETS

Socket L

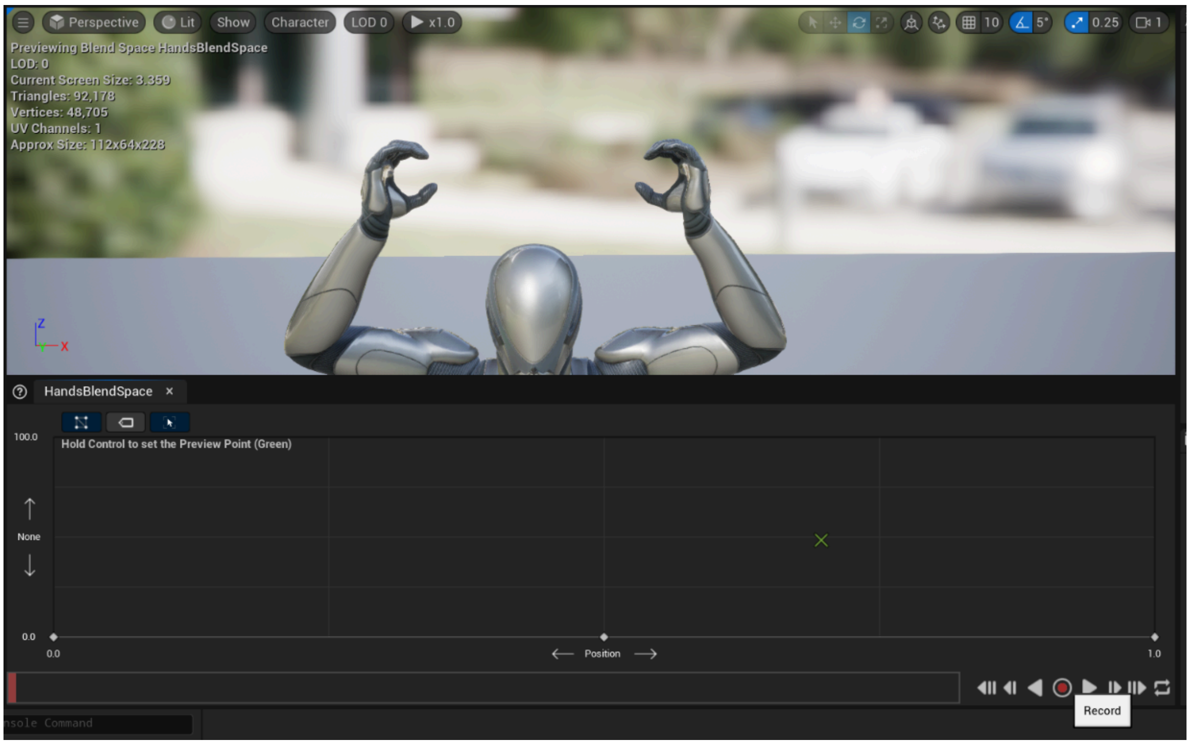
Socket R

2 sockets

Static Mesh Socket

Relative Location	25.0	70.75	-94.0
Relative Rotation	0.0	0.0	0.0
Relative Scale	1.0	1.0	1.0
Advanced			

When opening the Glider mesh model, it must have two Sockets, one for each hand. They can be created and named in the Socket Manager of the mesh. Place them where the hands make contact with the model. Make sure they use the same X and Z values but opposite Y values, like in the example Glider mesh.

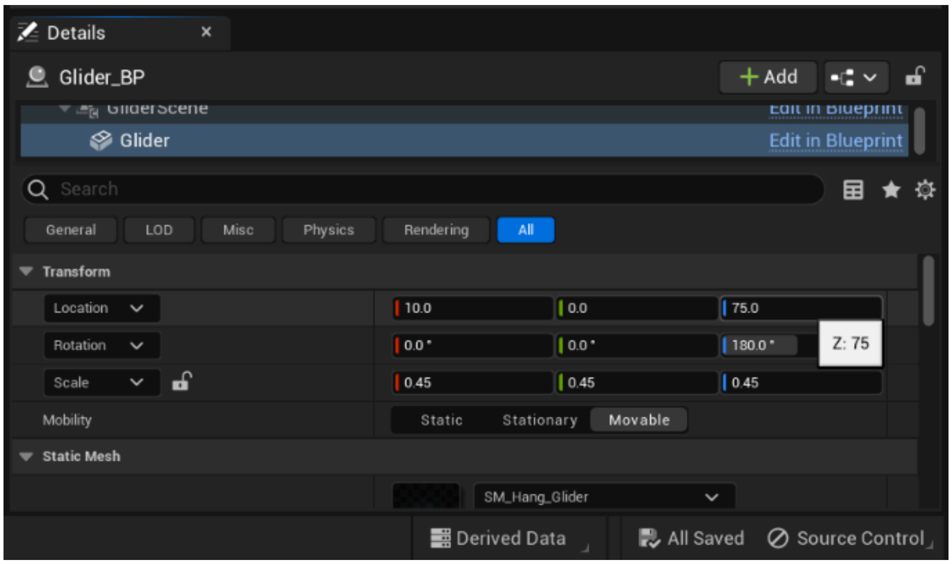

Previewing Blend Space HandsBlendSpace
LOD: 0
Current Screen Size: 3.359
Triangles: 92,178
Vertices: 48,705
UV Channels: 1
Approx Size: 112x64x228

Hold Control to set the Preview Point (Green)

Now, open the **HandsBlendSpace1D** inside the Animations ⇒ Version 2 folder. Notice that we can change the Transform of the Hands by moving the Blend value from 0 to 1. We use this Blend Space to get an appropriate hand animation that can be used with your Glider model. For example, if your Glider model is bigger and needs the hands to be more open, we can modify the **Position** float value.

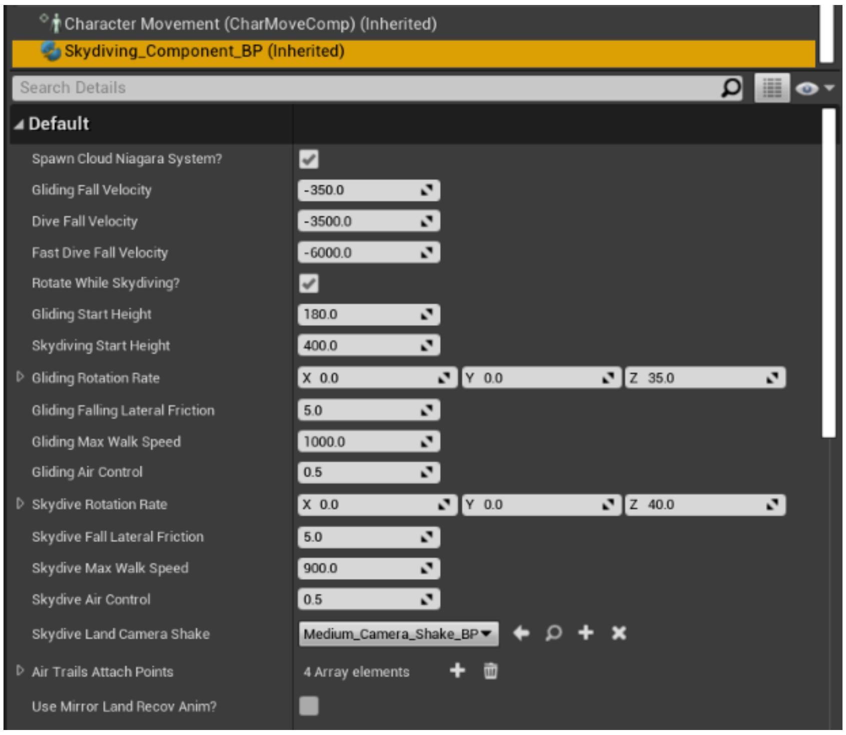
Go to the Animations ⇒ Animation Blueprint folder and open the Skydiving Anim Blueprint. Inside we can find the **Position** float variable. Change it to a value that suits the new Glider model, if used. Compile and save the blueprint.

- **Using The Component:**



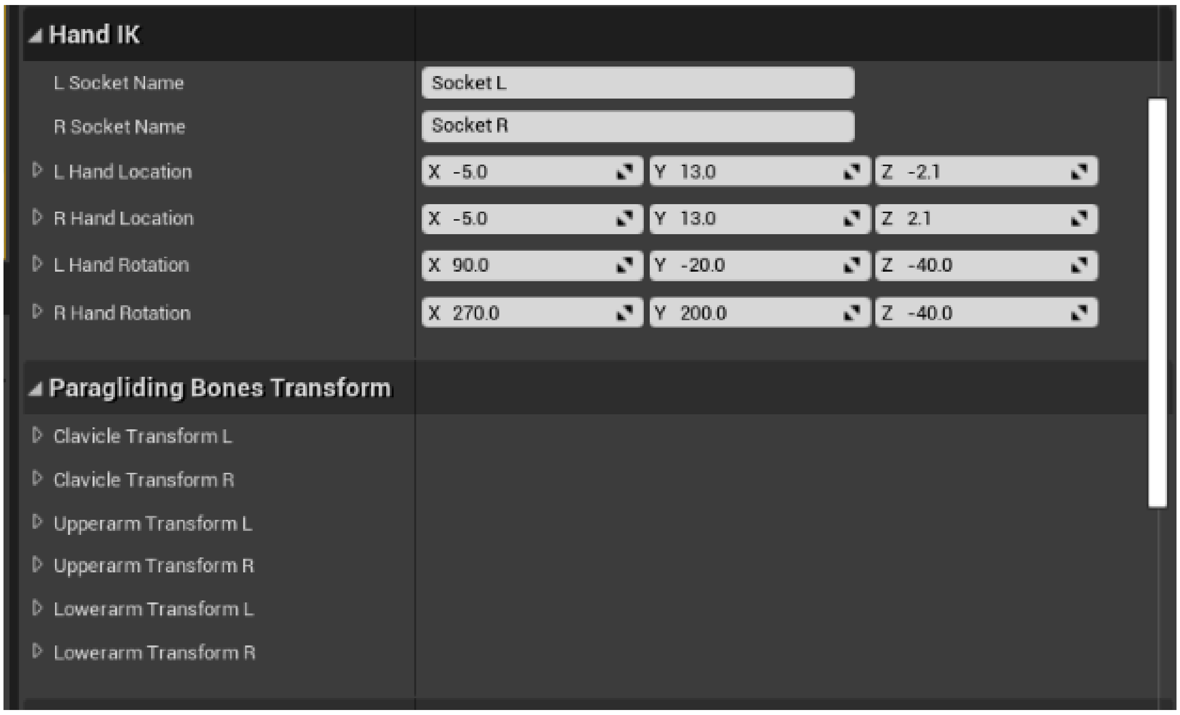
Transform
Location 10.0 0.0 75.0
Rotation 0.0 ° 0.0 ° 180.0 ° Z: 75
Scale 0.45 0.45 0.45
Mobility Static Stationary [✓ Movable]
Static Mesh
SM_Hang_Glider

First, notice that while Gliding, if the Glider mesh is placed too close or far away from the Character, the rotation of both arms will be incorrect. It can also happen if the Socket locations are placed too close or far from the Character. To fix this, adjust the **Location** of the Glider mesh (***not the GliderScene***) inside the Ver 2 Glider BP Actor placed in the level (not the glider actor world location but the location of the mesh inside it).


Character Movement (CharMoveComp) (Inherited)
Skydiving_Component_BP (Inherited)
Search Details

Default
Spawn Cloud Niagara System? ✓
Gliding Fall Velocity -350.0
Dive Fall Velocity -3500.0
Fast Dive Fall Velocity -6000.0
Rotate While Skydiving? ✓
Gliding Start Height 180.0
Skydiving Start Height 400.0
Gliding Rotation Rate X 0.0 Y 0.0 Z 35.0
Gliding Falling Lateral Friction 5.0
Gliding Max Walk Speed 1000.0
Gliding Air Control 0.5
Skydive Rotation Rate X 0.0 Y 0.0 Z 40.0
Skydive Fall Lateral Friction 5.0
Skydive Max Walk Speed 900.0
Skydive Air Control 0.5
Skydive Land Camera Shake Medium_Camera_Shake_BP
Air Trails Attach Points 4 Array elements
Use Mirror Land Recov Anim?

We can also select the Character in the level and adjust the Skydiving Component parameters. Keep in mind that you can hover over any variable with the mouse and read its description if you need.


Hand IK

L Socket Name	Socket L
R Socket Name	Socket R
L Hand Location	X -5.0 | Y 13.0 | Z -2.1
R Hand Location	X -5.0 | Y 13.0 | Z 2.1
L Hand Rotation	X 90.0 | Y -20.0 | Z -40.0
R Hand Rotation	X 270.0 | Y 200.0 | Z -40.0

Paragliding Bones Transform

- Clavicle Transform L
- Clavicle Transform R
- Upperarm Transform L
- Upperarm Transform R
- Lowerarm Transform L
- Lowerarm Transform R

Finally, the Hand IK location and rotation values, the Glider Socket names and other bones transform values can be modified here. First, if using another glider model, add the Right and Left Socket names used in your mesh. Then, adjust the Left and Right hands location and rotation by testing in the level. Getting the correct values can take some time.

The other bones transform values (clavicle, upper and lower arms) don't need to be adjusted immediately. It is possible to leave them as it is (0,0,0) and only change them if you want little adjustments to both arms while paragliding.

If you have any questions or need assistance, please ask me in the Questions section in the Product's Page at the Marketplace, I will respond to you as soon as I can.