

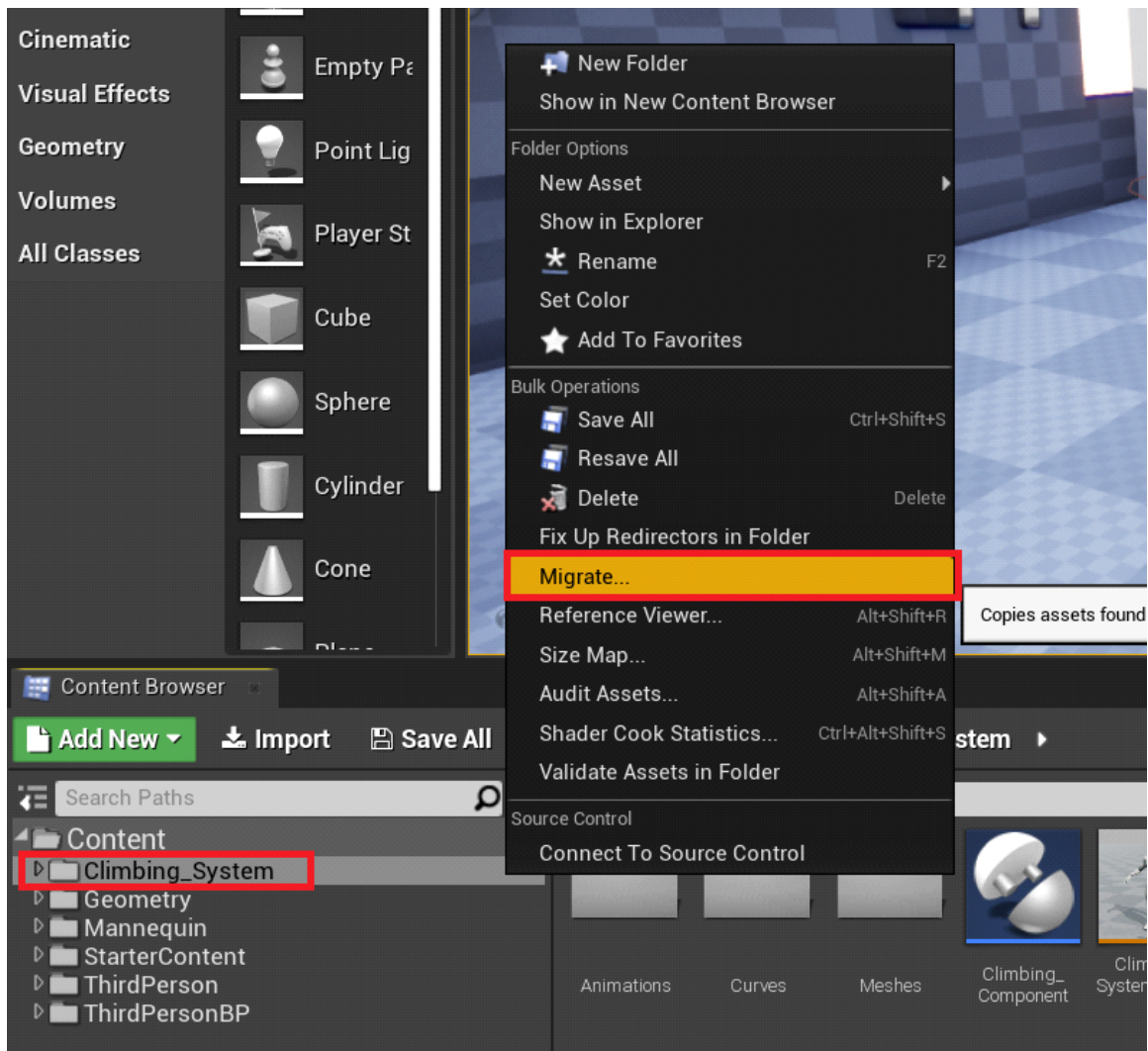
Ledge_Climbing_System

- 1, - Migrate climbing assets
- 2, - Modify Player BP
- 3, - Modify Player Anim BP
- 4, - Modify Player Skeleton:
- 5, - Hit Events
- 6, - Collision
- 7, - Make Object Climbable
- 8, - Climbing_Component

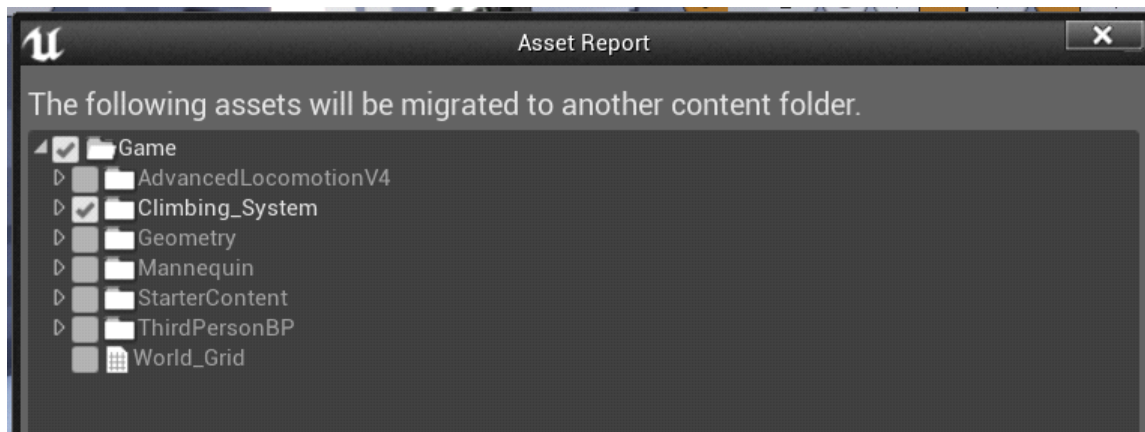
1, Migrate climbing assets:

Open the original Climbing System project.

Right click on Climbing_System folder and click Migrate...

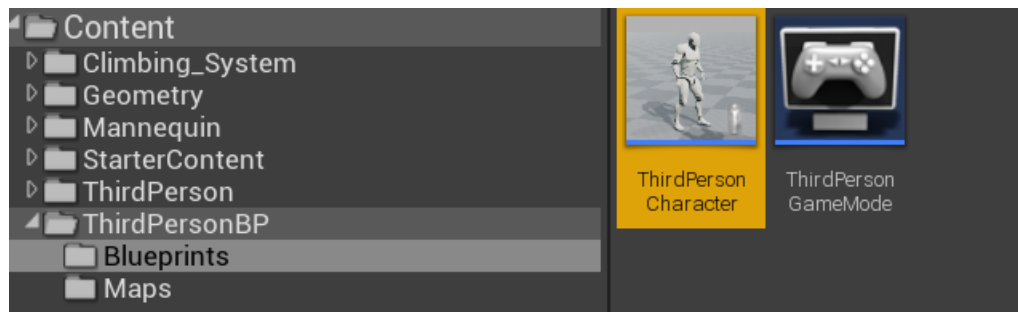


Select only Climbing_System folder and navigate to the target project file's content folder.



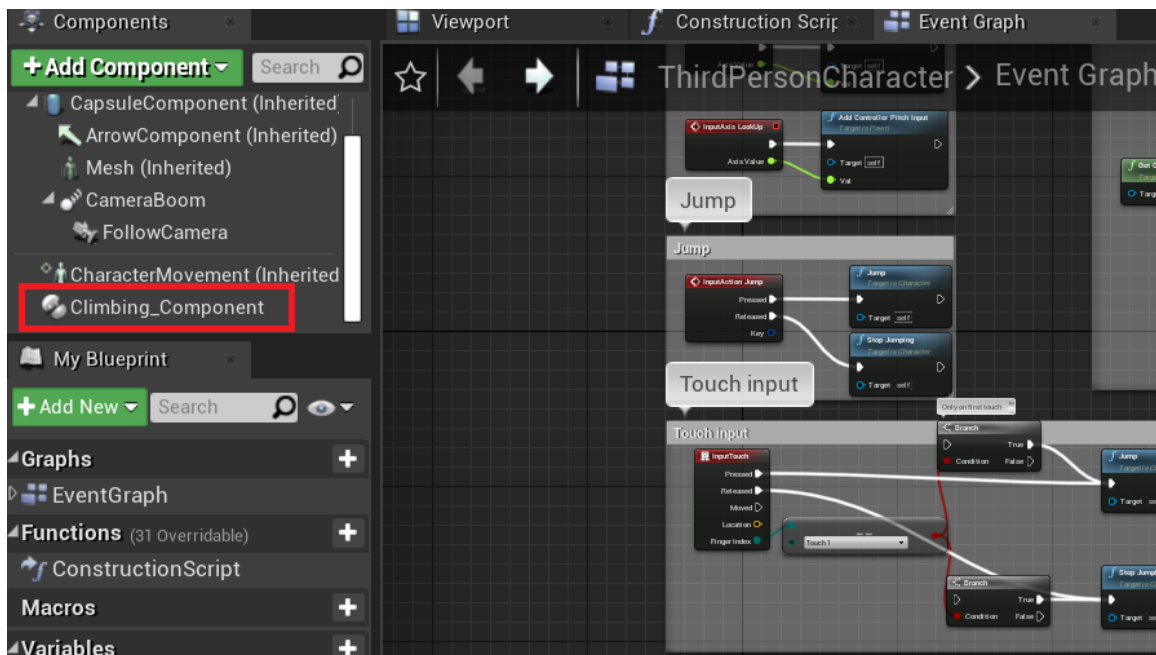
2, Modify Player BP:

In the original project ThirdPersonBP/Blueprints/ThirdPersonCharacter

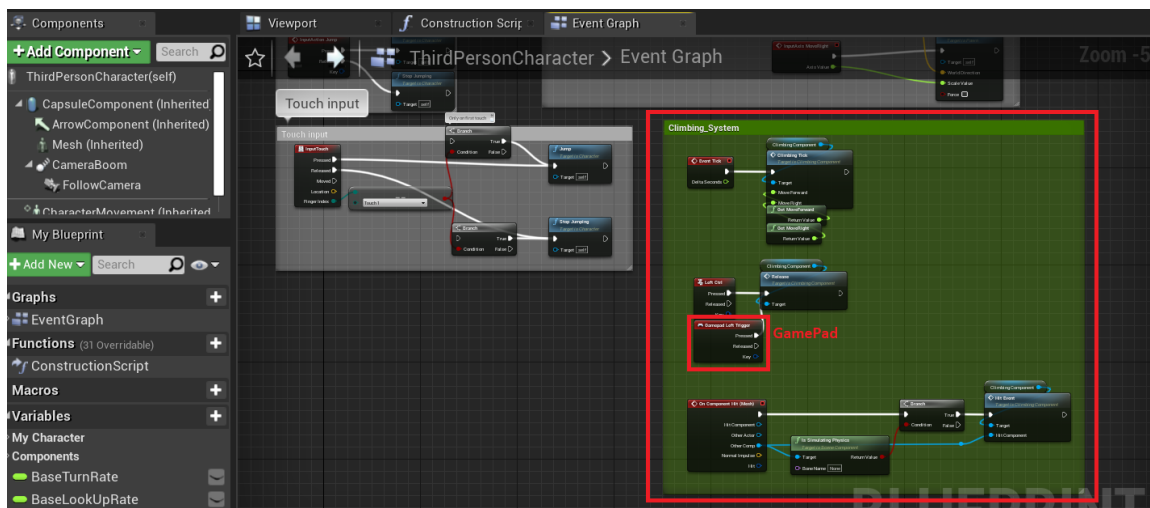


Open it.

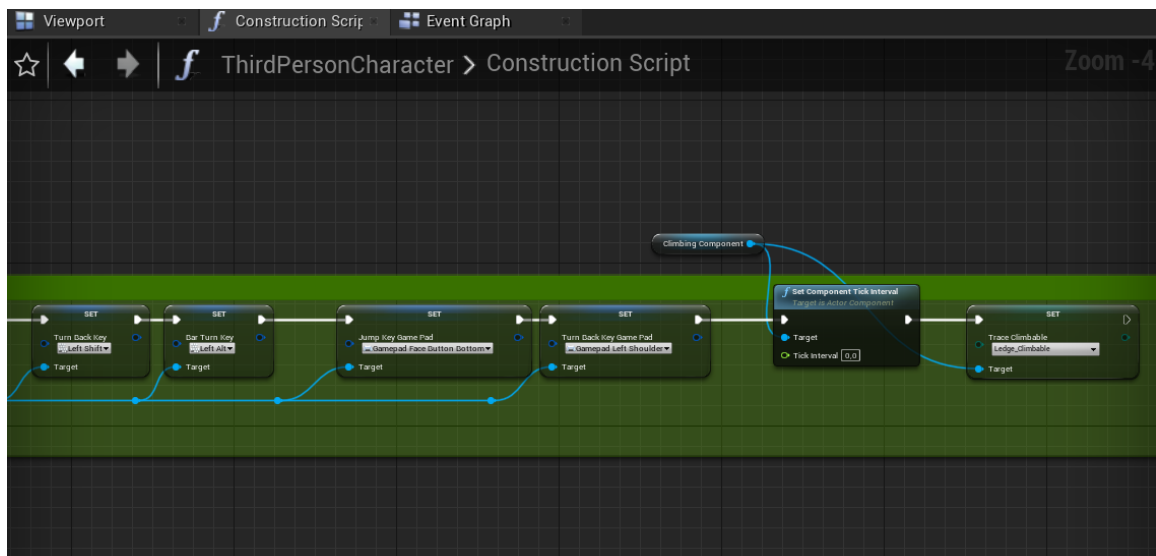
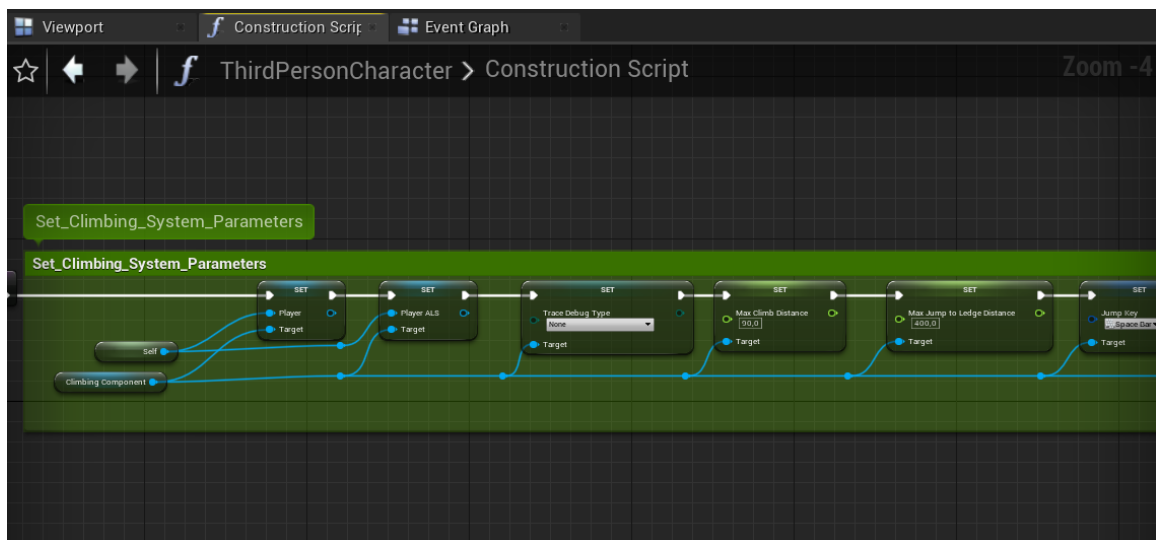
Add Climbing_Component.



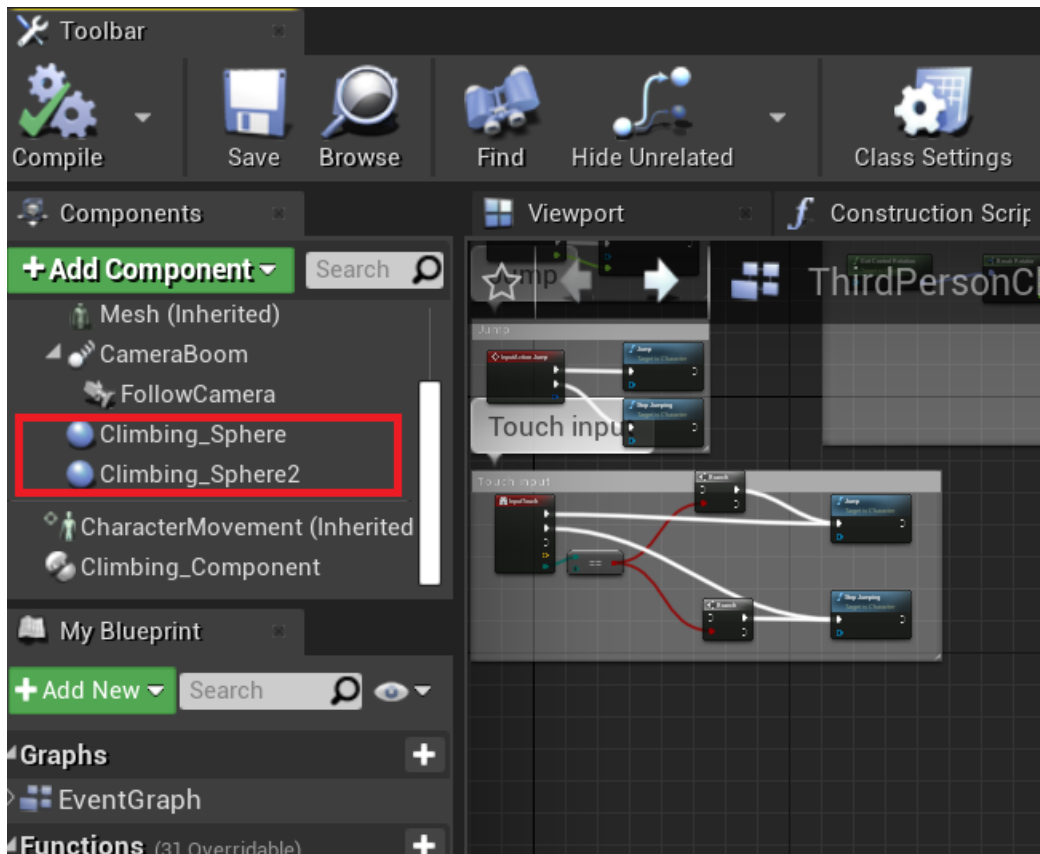
From the original (ledge climbing system project) character blueprint's event graph copy the Climbing nodes into the new blueprint.



Open the ConstructionScript in both player blueprints and copy the climbing nodes from the original project into the new one.



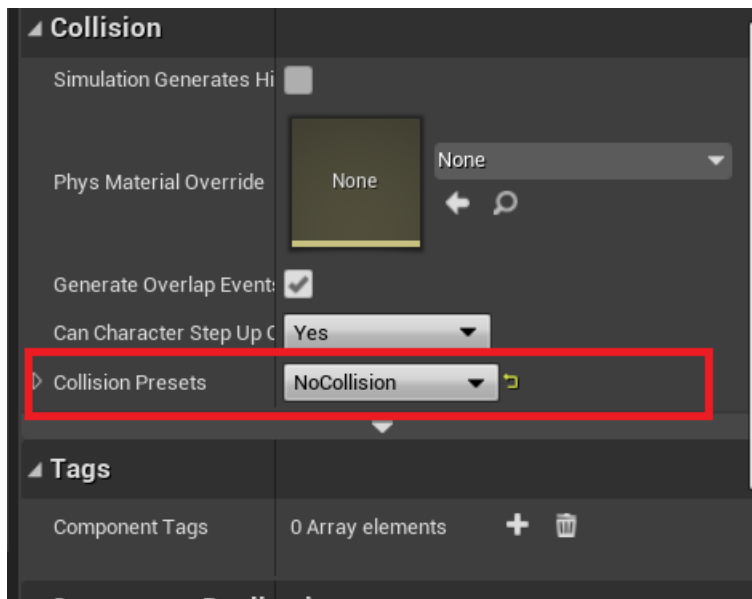
You need to add two Sphere Collisions.



Sphere radius: 5

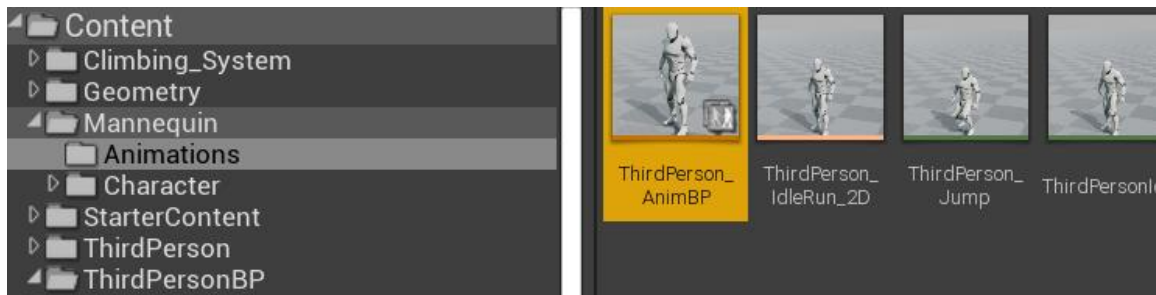


and there's no collision.



3, Modify Player Anim BP:

In the original project Mannequin/Animations/ThirdPerson_AnimBP



There's two options based on the system will be used in UE4 or UE5.

The climbing system has an Anim BP with Animation Notifications. This Anim Bp is linked in the main Anim BP of the player character.

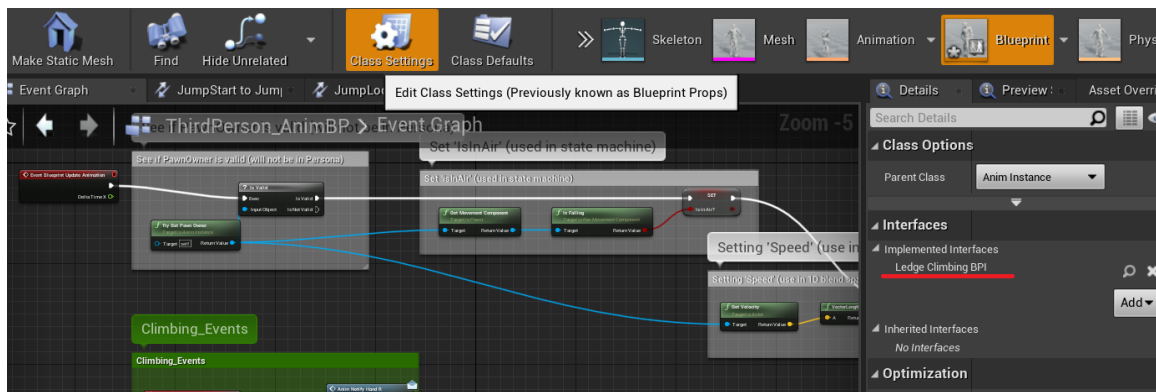
In UE5 Early Access version AnimNotifies don't fire in Linked AnimGraph (Anim Blueprints) only in the Main AnimGraph. So we need to call these parts in the main Anim BP through the Climbing_Interface.

In UE5, the climbing system Anim BP the Animation Notifications parts have to be called from the Main Anim BP through Climbing Interface Events.

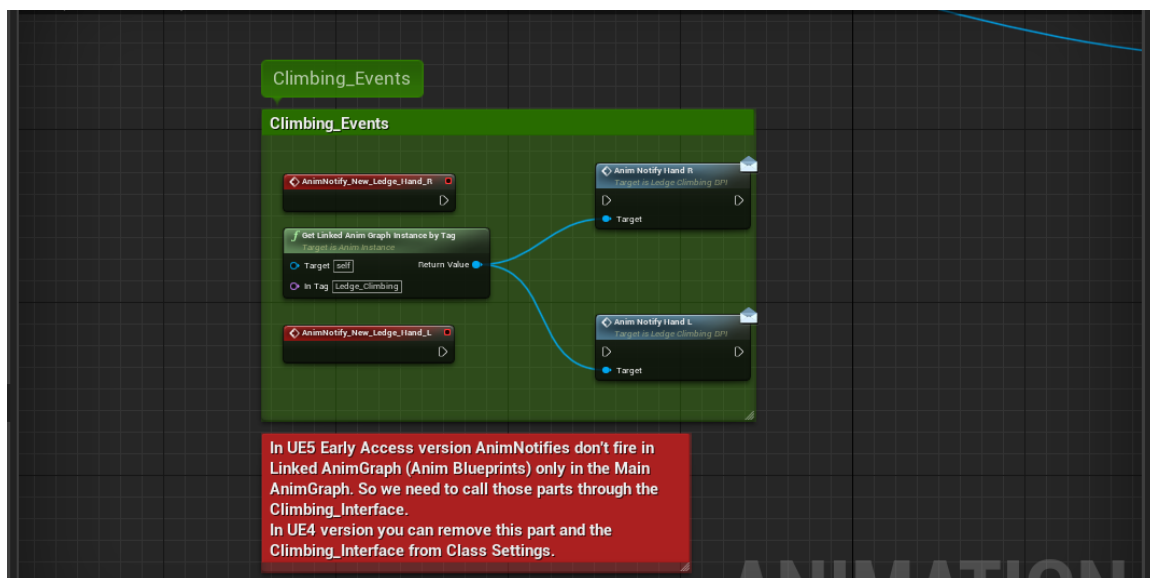
UE5 Option:

Open the Class settings in the new animation blueprint you want to use with Ledge Climbing.

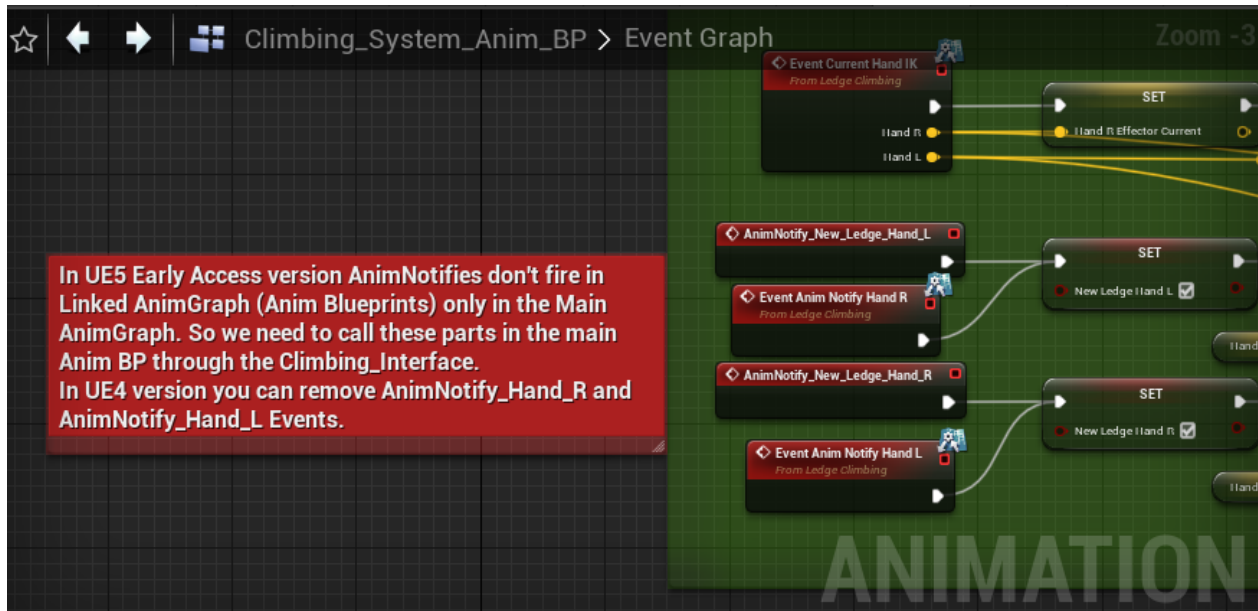
Open class settings and add Ledge_Climbing interface.



From the Original Animation Blueprint's Event Graph copy the Climbing Events into the new animation blueprint's Event Graph and connect the two events to the interface calls.



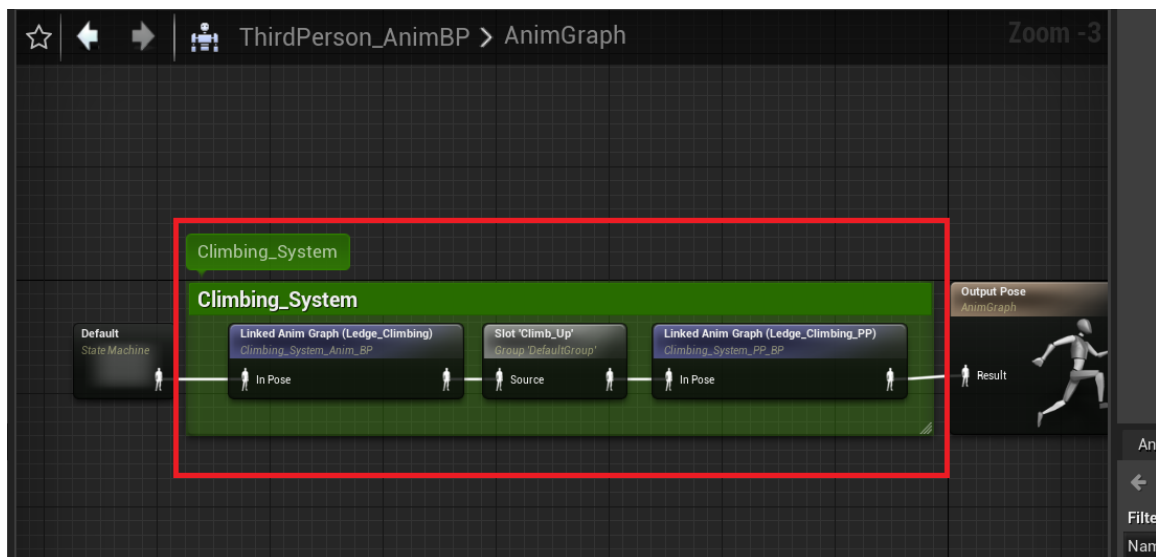
This is how the Animation Notification part in the Climbing System Anim Bp looks like:



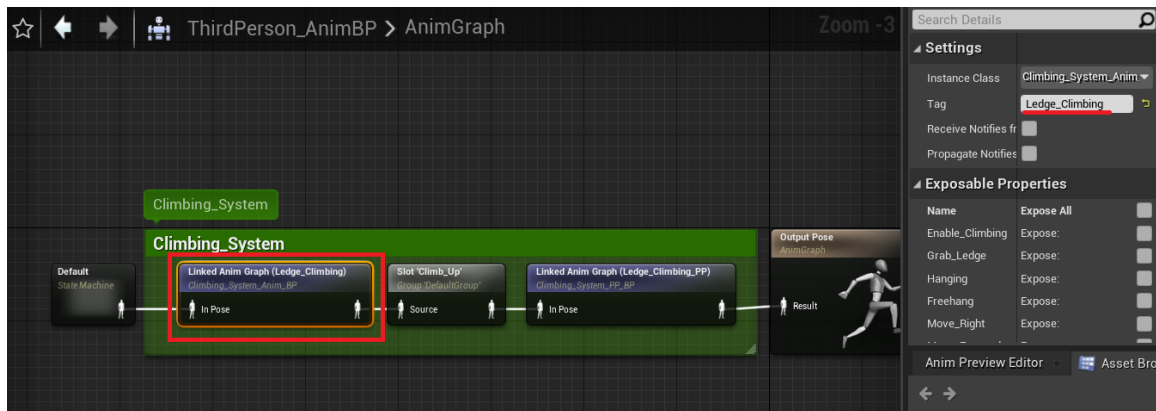
UE4 Option: You don't have to add the Ledge Climbing interface and the Climbing events into the EventGraph.

Do with Both Version:

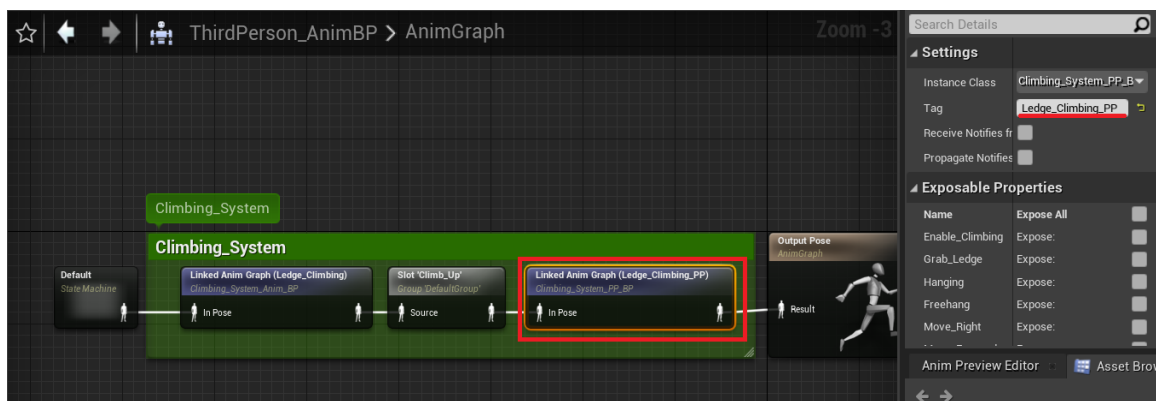
In Anim Graph Copy the climbing nodes from original to the new anim blueprint.



Be sure the linked anim graph has Ledge_Climbing tag.



and the other linked anim graph has Ledge_Climbing_PP tag.



4, Modify Player Skeleton:

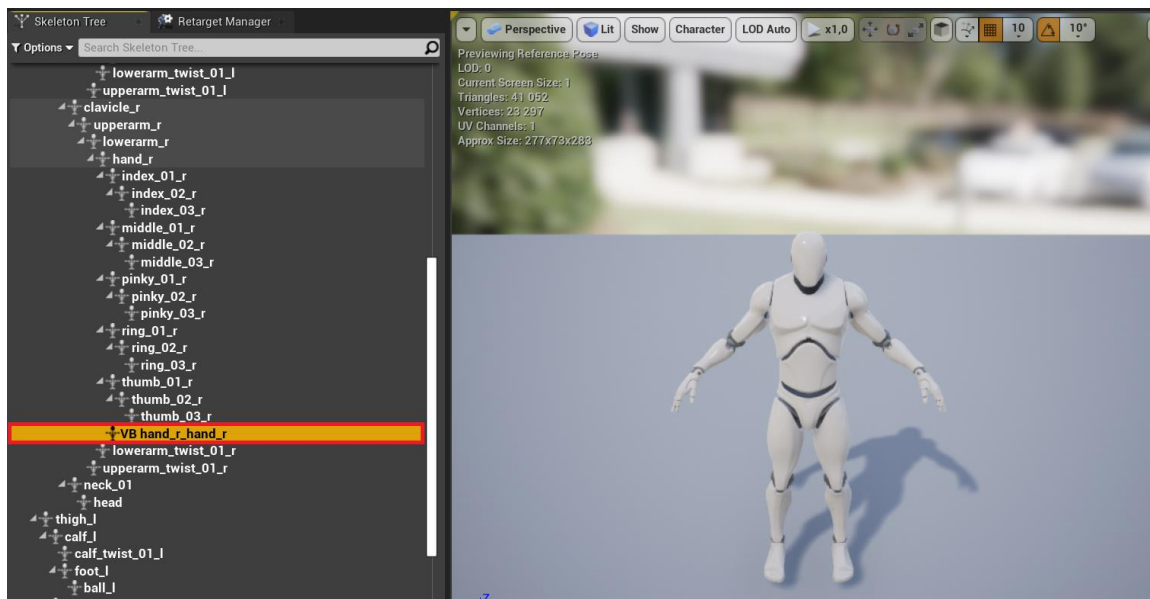
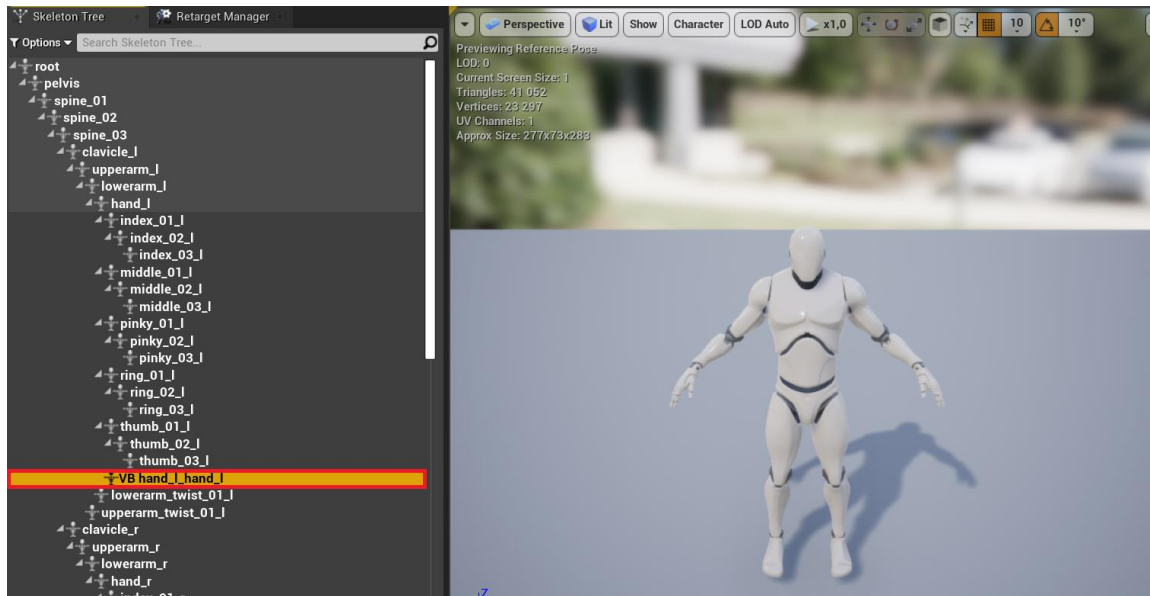
Add Virtual bones.

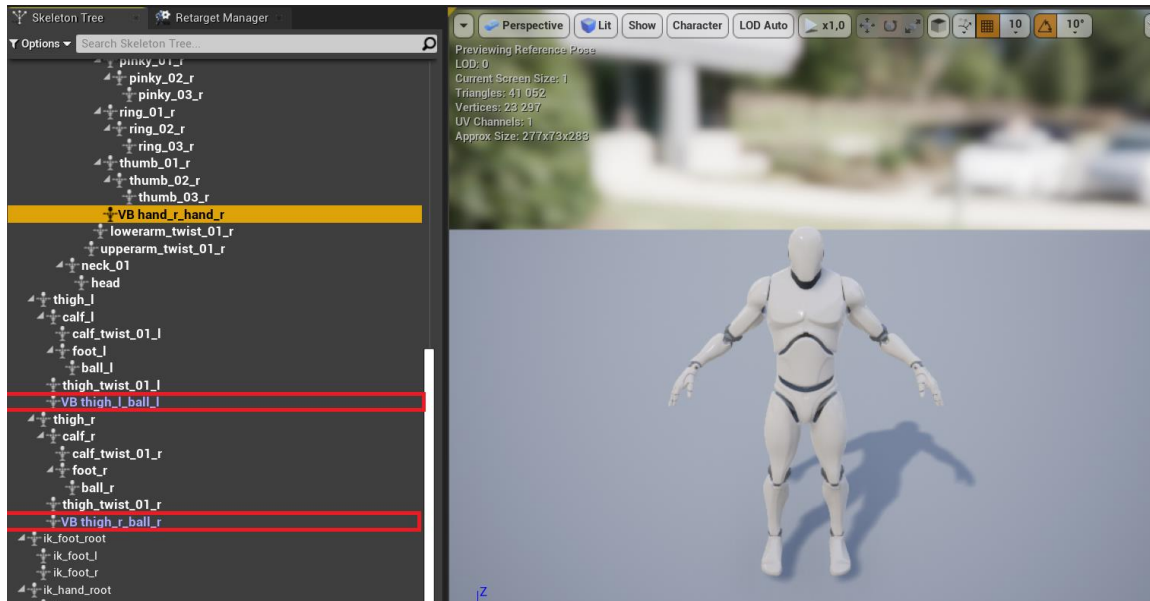
Hand_L - Hand_L

Hand_R - Hand_R

Thigh_L - Ball_L

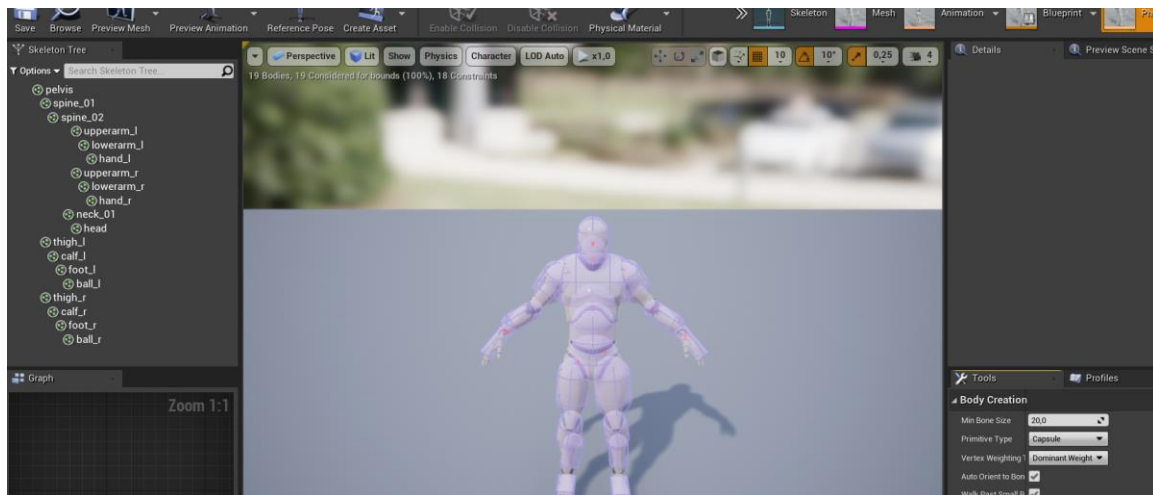
Thigh_R - Ball_R



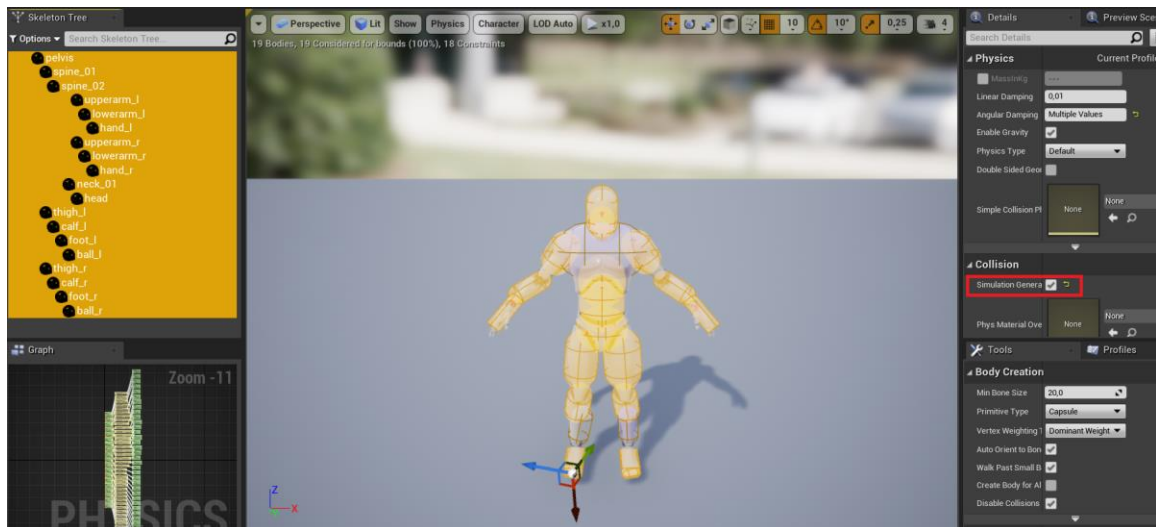


5, Hit Events:

Open the SK_Mannequin_PhysicsAsset.



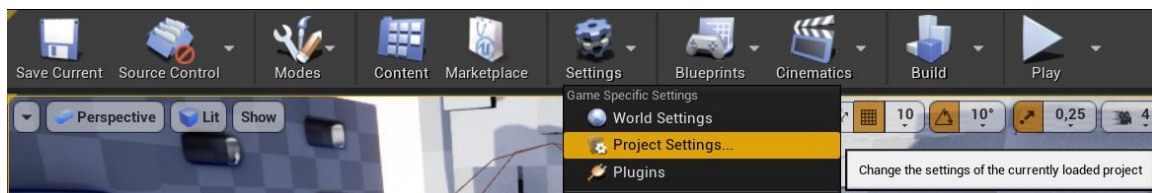
Select all the bones you want to be simulated. Set 'Simulation Generates Hit Event'. These bones will detect hits and fire an Event.



6, Collision:

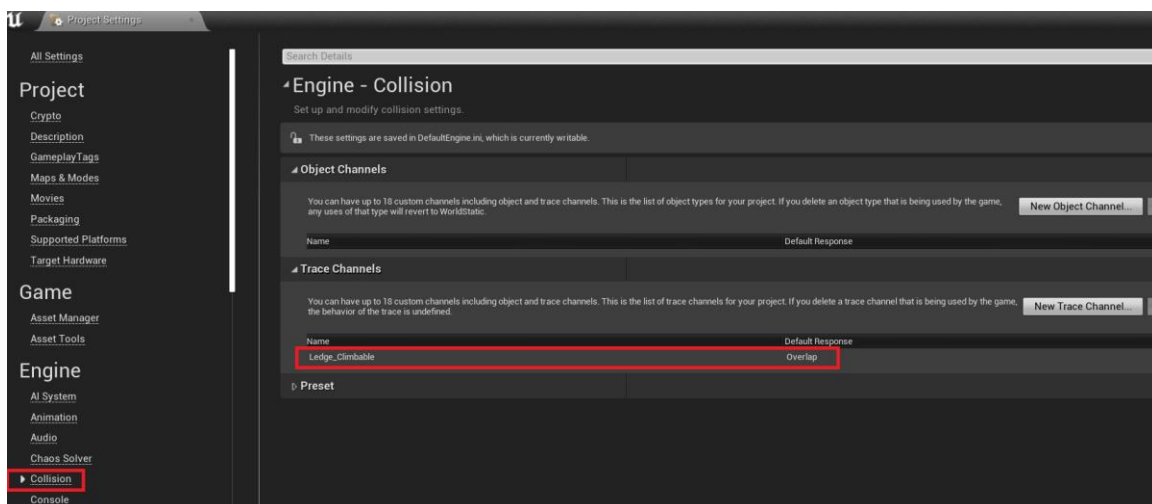
Select all the bones you want

In the new project, Open Project Settings.



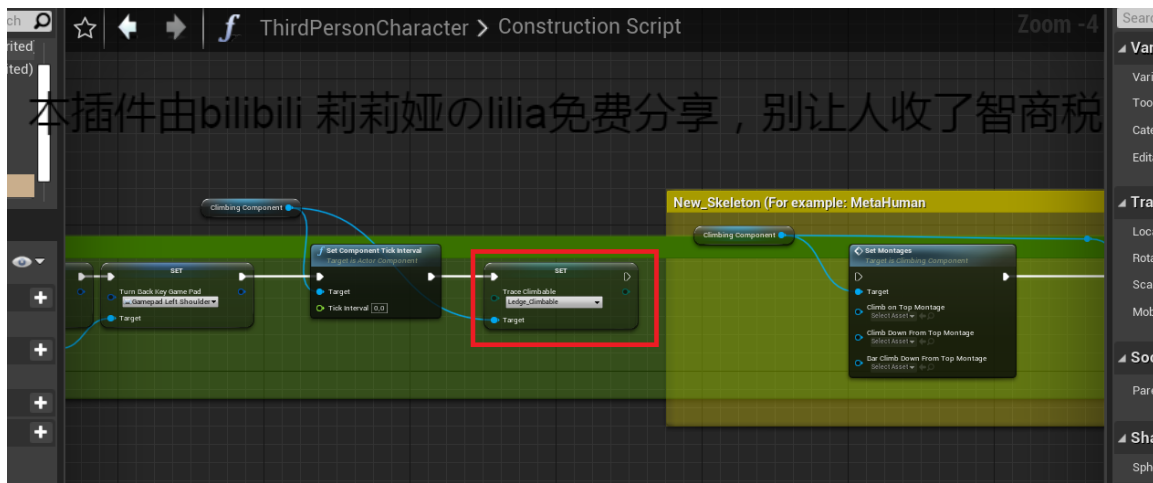
Under Collision, add a new Trace Channel.

Set the Default Response to Overlap.



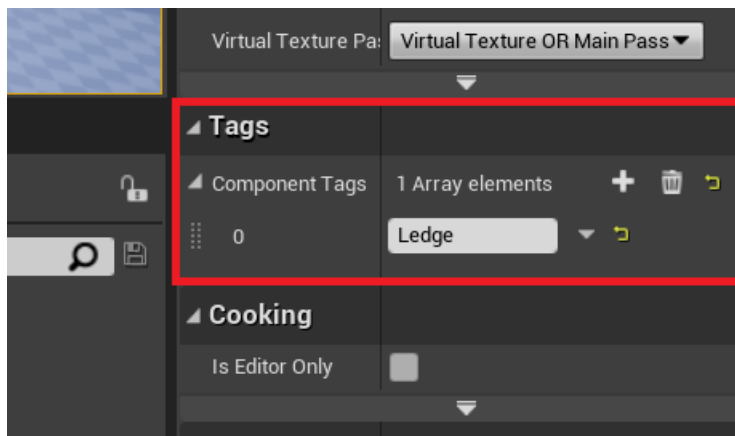
Open the ThirdPersonCharacter. Under ConstructionScript, Climbing System Nodes and set the

Trace_Climbable node to the newly added trace channel.

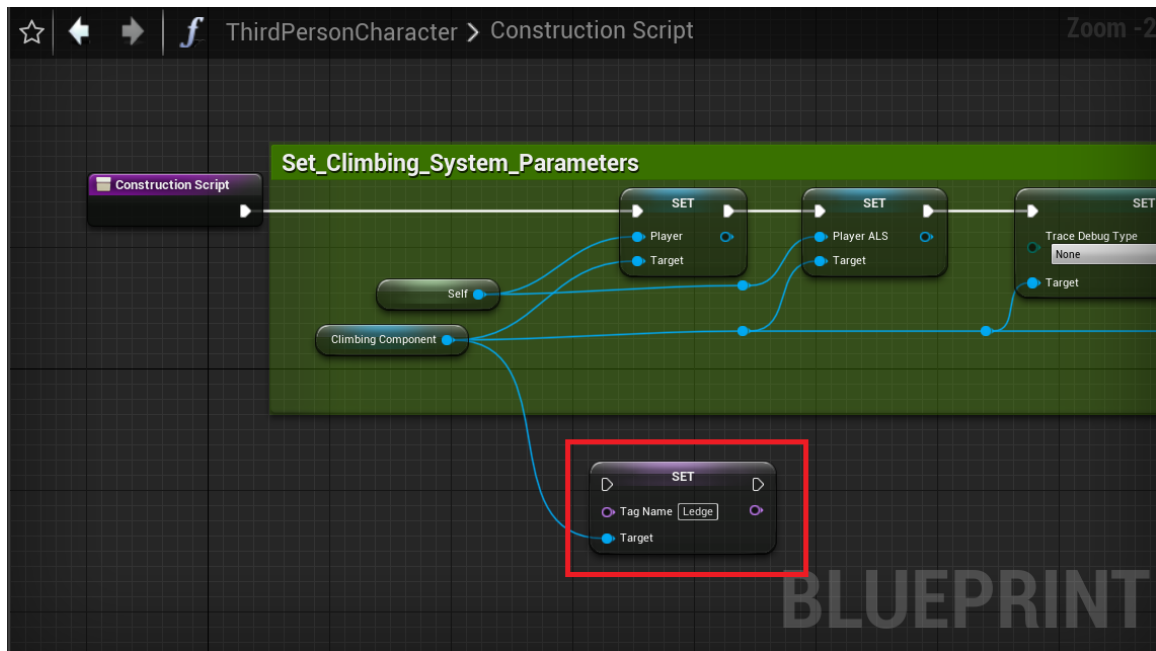


7, Make Object Climbable:

Add 'Ledge' tag to the object.



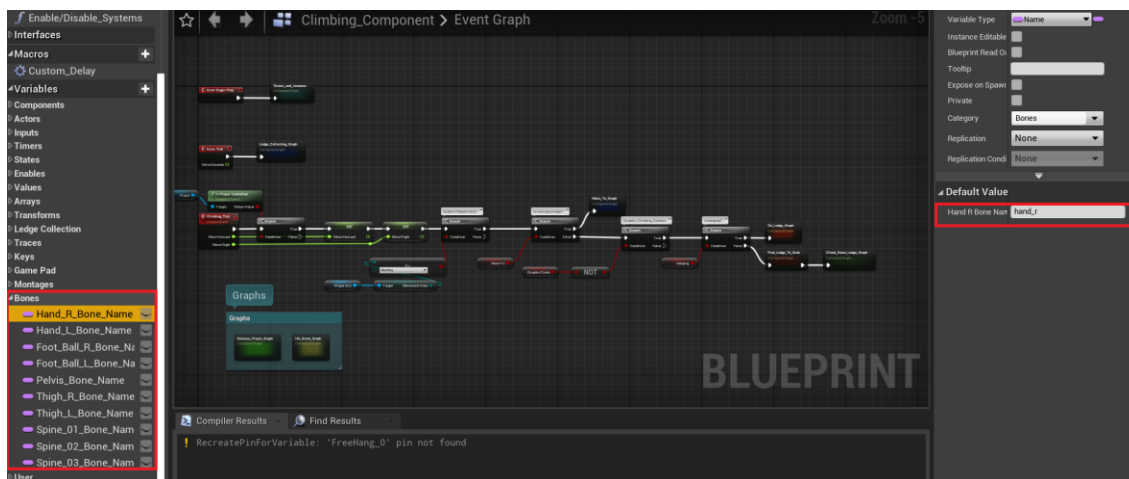
Or you can change the tag name in character BP ConstructionScript.



8, Climbing Component:

User_Variables:

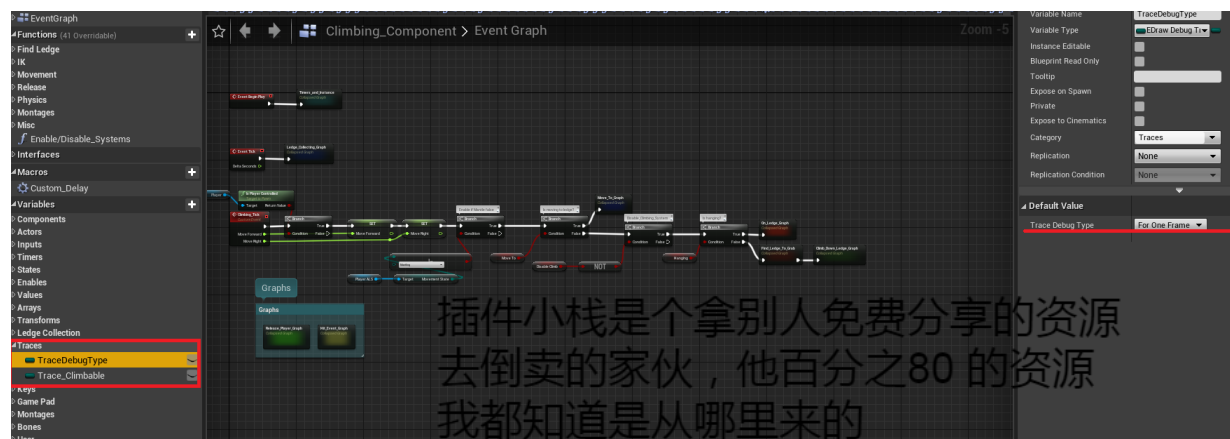
Bones: You can set the name of the bones according to your player's skeleton.



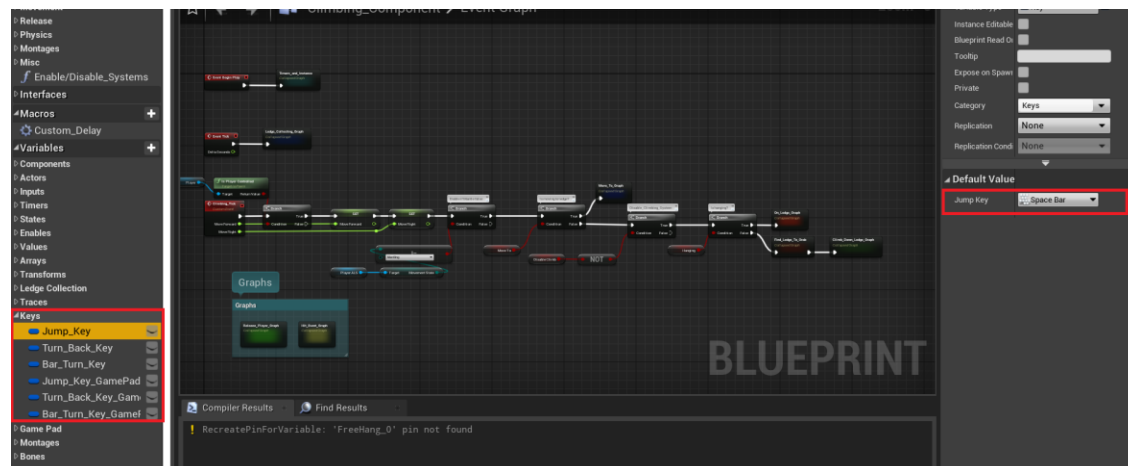
Traces:

[TraceDebugType]: You can debug the traces and other stuffs.

[Trace_Climbable]: The trace channel you can use to detect climbable objects.



Keys:



User:



[Slope_Limit]: The ledge angle the player can grab. If 0: Z=1(Flat), if 1: Z=0.

[Hand_Distance]: The sideways hand offset value for the hands.

[Max_Climb_Distance]: Above this value, the player will jump and not climb to the next ledge.

[Tag_Name]: The object tag name to make the object climbable. Default: 'Ledge'.

[Hand_Movement_Offset]: Hand offset from the hand grab points. It's usefull for different characters with different hand size and arm length. Used when moving (Shimmy) on ledge.

[Hand_Idle_Offset]: Hand offset from the hand grab points. It's usefull for different characters with different hand size. Used when hanging or climbing on ledge.

[Leg_IK_Depth]: The forward length of the leg IK trace.

[Capsule_Radius_Multiplier]: Set the player capsule size while climbing. Default value is 1. If 0.5, the capsule will be the half of the original size. It's good if you want the player to be able to grab a ledge when there's something closer to the player under the grab point and there's not enough room for the player capsule.

[Use_CCD]: In the ThirdPersonCharacter, at the collision settings of the mesh, under the extra options you can nable the continous collision detection for the player mesh. If it's enabled by default, the Use_CCD flag can be unchecked here. Otherwise the climbing component will enable it when you climb and disable it when you release the ledge.

[Capsule_Position]: The player capsule offset from the ledge point.

[Freehang_Check_Offset]: X - sideways, Y - forward, Z - upward. variable for traces to check when should the character switch to freehang.

[Use_Leg_Correction]: If it's enabled, the legs will be pulled closer to character when the legs can't make contact to the wall but it's not switched to freehang yet.

Extra: **this plugin is free asset ,share by bilibili lilia**
<https://b23.tv/pTrWgqu>

[Is_Active] flag: Usefull when you want to check outside the climbing component that the climbing is active or not.

[Enable/Disable_Systems]: You can add nodes of other components (Player or Player_ALS variable) to disable and enable systems while climbing. True branch for disable, false branch for reenable systems. This function is already placed in the climbing component.

