

Thank you for your purchase of Noitom Hi5 VR gloves. The Hi5 Unreal Plugin provides all the necessary functions and interfaces to the Noitom Hi5 gloves. There is no more software needed to drive the Hi5 gloves.

**Version**

* Hi5\_Unreal\_Plugin\_0\_9\_8\_655\_1

**System Requirements**

* Unreal 4.14+
* Visual Studio 2015
* Windows 7+

**Calibration**

Before starting your development, first make sure your glove is calibrated. Download the Hi5 Sample on <https://hi5vrglove.com/>.

1. To use the Hi5 glove, firstly please confirm all the preparation steps below:
   1. The HTC VIVE system is running in a clean environment, without any optical reflection and other IR interference.
   2. The SteamVR system is running correctly.
   3. Attach the VIVE controller or VIVE tracker on each of your Hi5 glove.
   4. Turn on the attached controllers or trackers.
   5. Insert the Hi5 dongle to the USB port of your computer/ HMD.
   6. Turn on the Hi5 gloves.
2. Open the Hi5 calibration executable file.
3. Follow the instructions to do the calibration. The full calibration procedure includes two steps. The first step is B-pose calibration, if you successfully do B-pose, you will see your hands in VR.
4. The second step is P-pose calibration, which means pinch pose calibration. This allows you to achieve “pinch (finger touch)” in VR.
5. Once you finished the calibration procedure, it will save the calibration information to the default folder. Next time when you start any of your Hi5 projects, it will load the calibration files directly.

**Note:**

1. If you reset the VIVE lighthouse, or change other controllers/ trackers, you need to recalibrate your gloves.
2. All of the Hi5 Sample resources are provided in the Hi5\_Mix Sample.

**Prepare for Development**

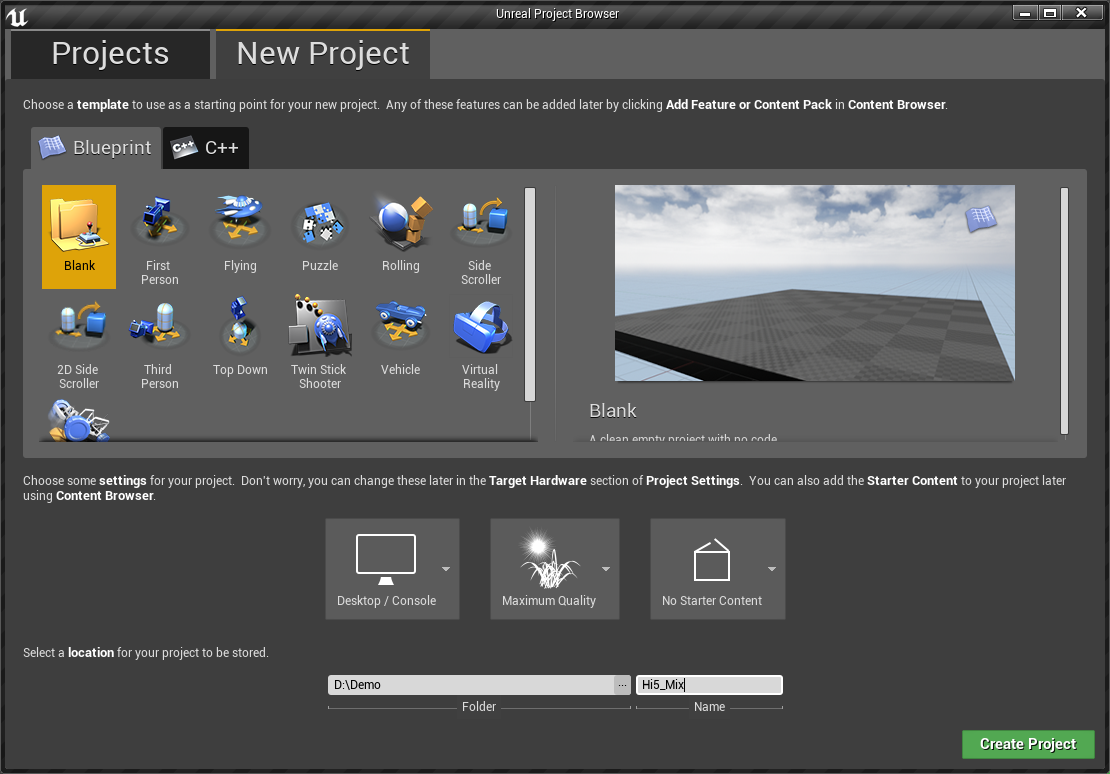
1. Please make sure you already correctly setup the HTC VIVE environment. Follow the HTC VIVE setup procedure as below.

<https://www.vive.com/us/support/category_howto/settings.html>

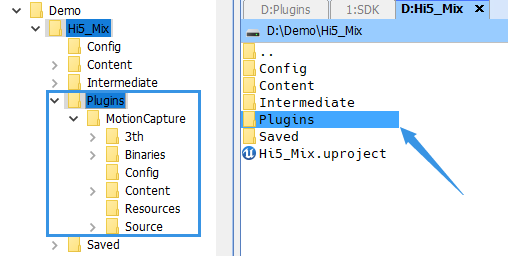
**Usage of the Plugin(For UE4.20, please read the end of the document firstly.)**

**Create Demo Project**

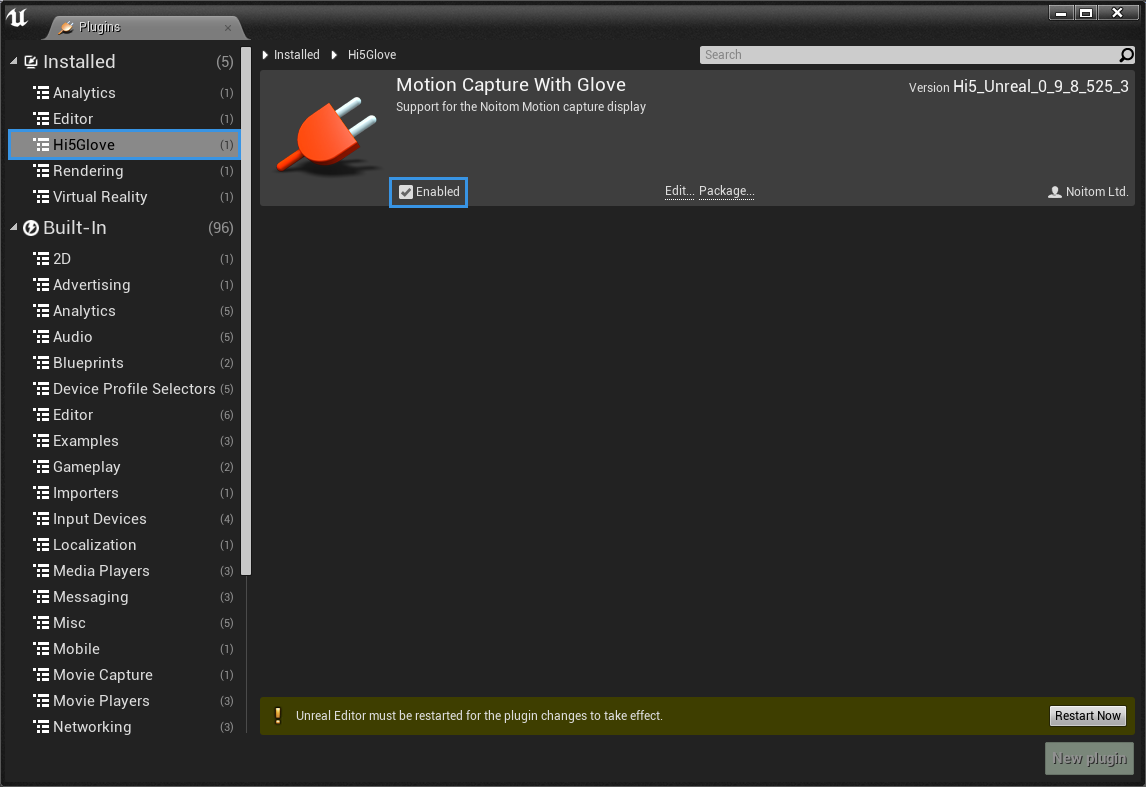
Create a Blank Blueprint project, name it “Hi5\_Mix”: ↓



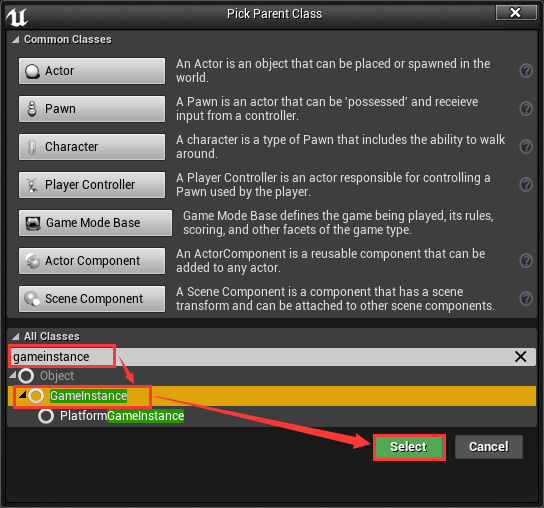
Then copy MotionCapture plugin to this project’s Plugins folder: ↓



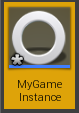
Close and Reopen the project, locate menu item: Editor -> Plugins, check if the MotionCapture plugin is listed and enabled, as follow:



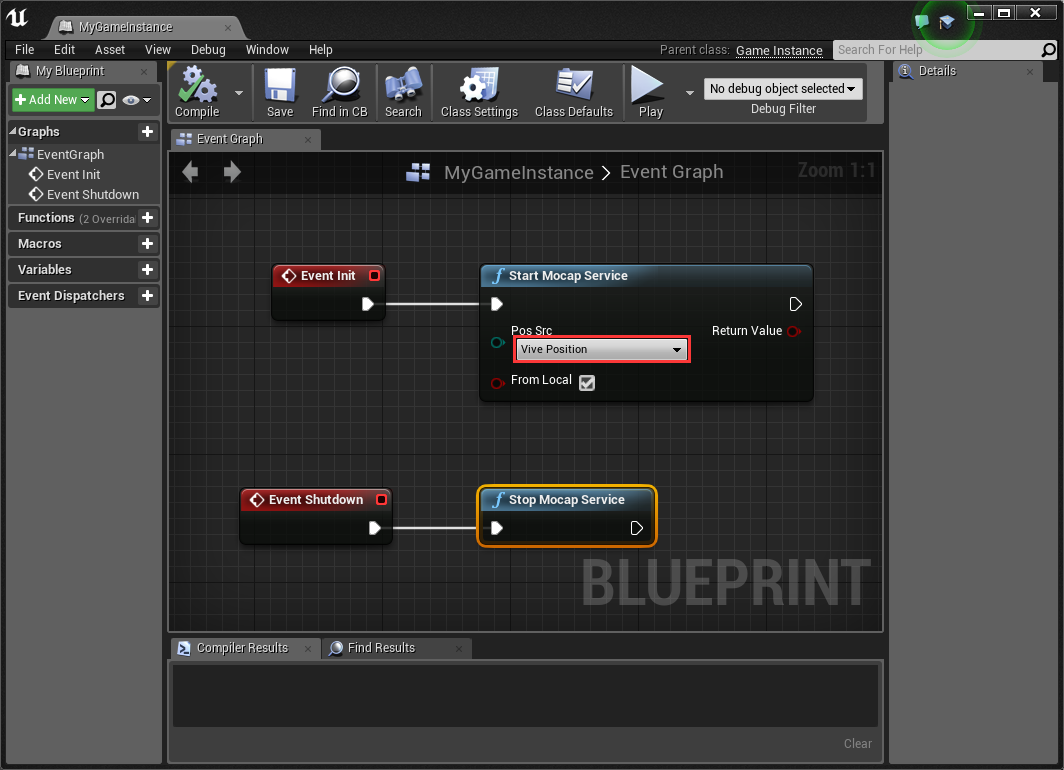
**Create GameInstance**



Create a GameInstance Blueprint and name it “MyGameInstance”, as follow:

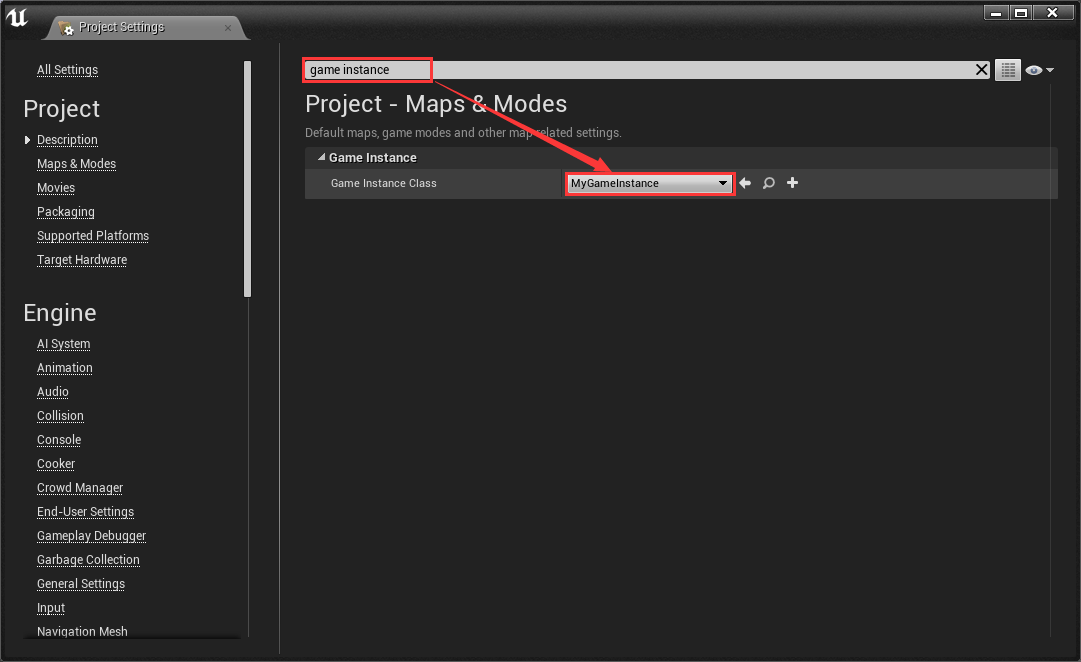


Double click the icon to edit MyGameInstace as follows:

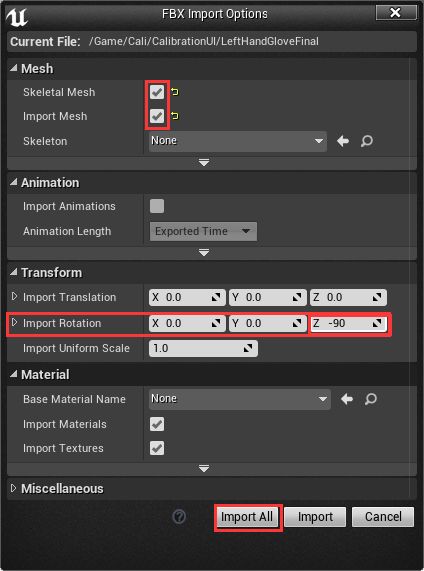


Note: The current version plugin only support HTC Vive Optical system.

After finished MyGameInstance, locate UEditor menu item: Edit->Project Settings, setting as follow:

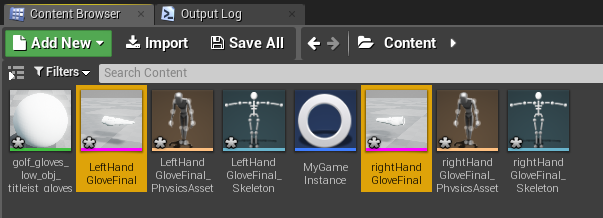


**Import Hands Models**



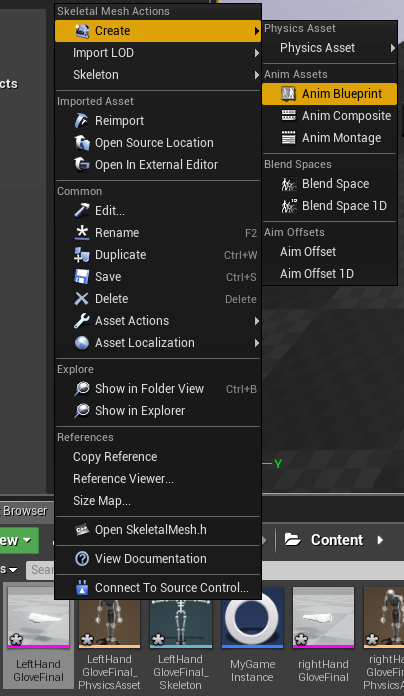
Note: the [Import Rotation] option, the Z is set to -90 degrees. This is because the imported hand models’ initial orientation doesn’t satisfy the requirement, So we modify the [Import Rotation] to fix the problem. The requirement of hand model’s initial orientation is clear: MotionCapture plugin defines positive X-axis of Unreal is forward, after hand models imported into Unreal, the left hand back should point to up(positive Z-axis in Unreal Engine), the left hand fingertips should point to negative Y-axis, the right hand back should point to up(positive Z-axis), the right hand fingertips should point to positive Y-axis. If the imported models not satisfy this requirement, you should reimport these models with correct orientation by modifying [Transform] options in [Import Options] dialog.

In Content Browser we can see the imported hand models: LeftHandGloveFinal, RightHandGloveFinal(The name is related to model file.)

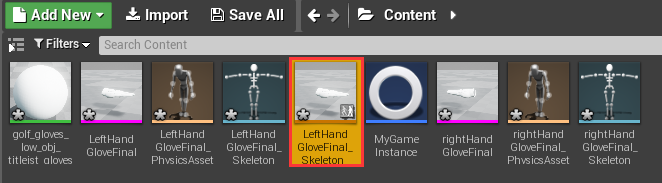


**Create Animation Blueprint**

We right-click on LeftHandGloveFinal mesh icon, then pick the [Animation Blueprint] popup menu item:

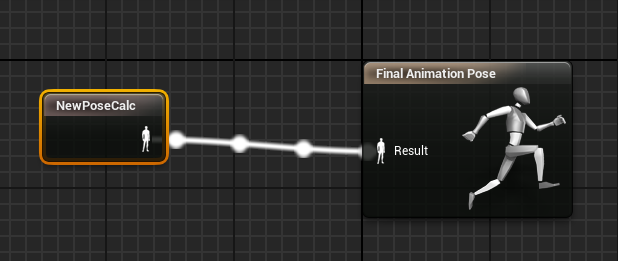


In Content Browser, we can see the newly created **LeftHandGloveFinal\_Skeleton\_AnimBlueprint:**



Double-click the new Animation Blueprint to edit it.

In **Anim Graph**, right-click to create a **NewPoseCalc** Node, then link the output of **NewPoseCalc** Node to the input Final Animation Pose Node. After all done, compile **Anim Graph:**



**Skeleton Retargeting**

MotionCapture plugin defines a standard skeleton hierarchy constructed by 42 bones. The names of these bones list as follows:

RightForeArm

RightHand

RightHandThumb1

RightHandThumb2

RightHandThumb3

RightInHandIndex

RightInHandIndex1

RightInHandIndex2

RightInHandIndex3

RightInHandMiddle

RightInHandMiddle1

RightInHandMiddle2

RightInHandMiddle3

RightInHandRing

RightInHandRing1

RightInHandRing2

RightInHandRing3

RightInHandPinky

RightInHandPinky1

RightInHandPinky2

RightInHandPinky3

LeftForeArm

LeftHand

LeftHandThumb1

LeftHandThumb2

LeftHandThumb3

LeftInHandIndex

LeftInHandIndex1

LeftInHandIndex2

LeftInHandIndex3

LeftInHandMiddle

LeftInHandMiddle1

LeftInHandMiddle2

LeftInHandMiddle3

LeftInHandRing

LeftInHandRing1

LeftInHandRing2

LeftInHandRing3

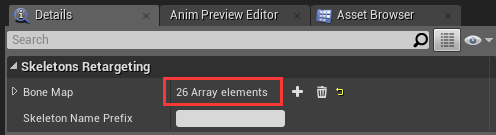
LeftInHandPinky

LeftInHandPinky1

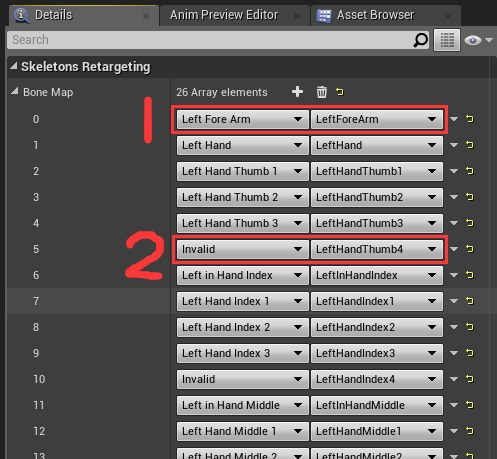
LeftInHandPinky2

LeftInHandPinky3

If the name of a bone in model file is in the list above, we say the bone is a standard bone. If a model file contains all standard bones, we say the model file is standard. If you import a non-standard hand model, you may need to do a Skeleton Retargeting to meet the plugin expected. In **Anim Graph**, select the newly created **NewPoseCalc** Node, in Details window there will display some options about Skeleton Retargeting as follow:

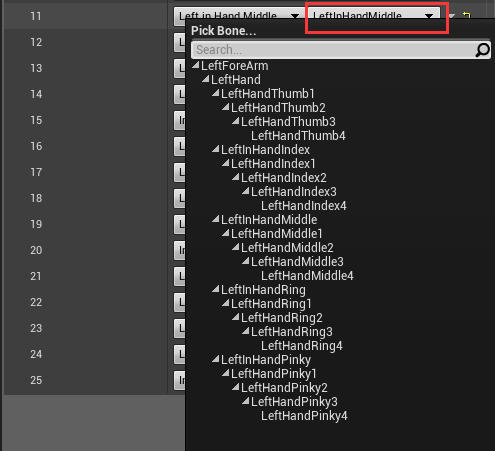
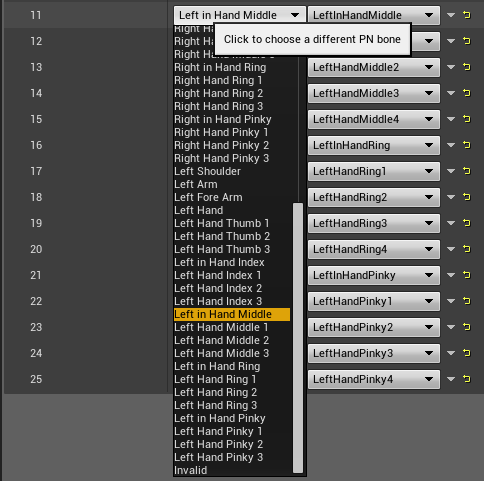


We can expand [Bone Map] to modify Skeleton Retargeting:

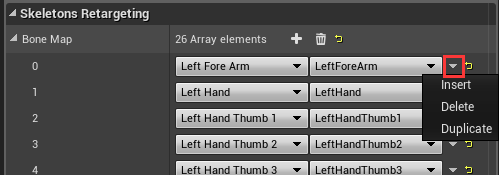


The left column is bone’s standard name defined in MotionCapture plugin. The right column is bone’s name defined in hand model file. If a bone name in model file is standard, then the left and right column will be automatically mapped correctly.(As the above picture showed: Red No.1). If a bone name in model file is non-standard, then the bone will be mapped to Invalid(As the above picture showed: Red No.2), and this bone will stay static relative to its parent bone. If all the names of bones in model have prefix, you need to fill the prefix into [Skeleton Name Prefix] edit box, then press enter key.

We can manually modify the corresponding relationship about Skeleton Retargeting:



We also can add or delete a bone mapping:

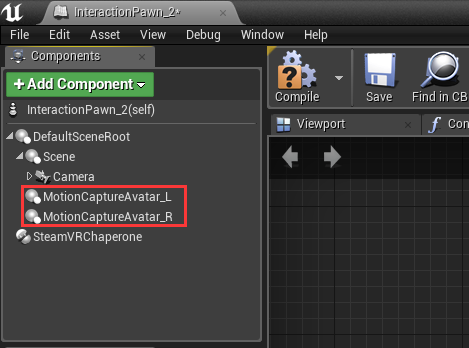


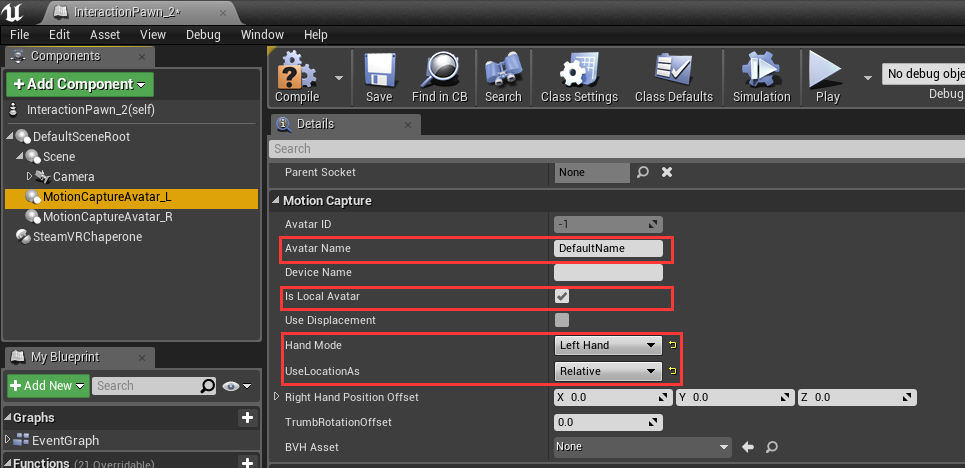
Note: The bone name in model file must be unique.

Do the same work for Right Hand.

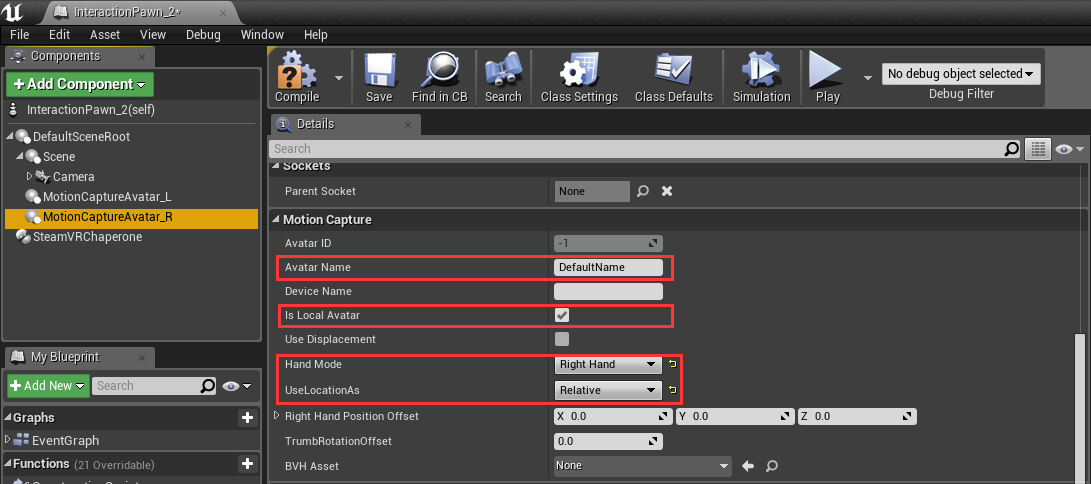
**Modify Pawn**

Add two **MotionCaptureAvatar** Components to **Pawn**, then rename them to MotionCaptureAvatar\_L and MotionCaptureAvatar\_R.



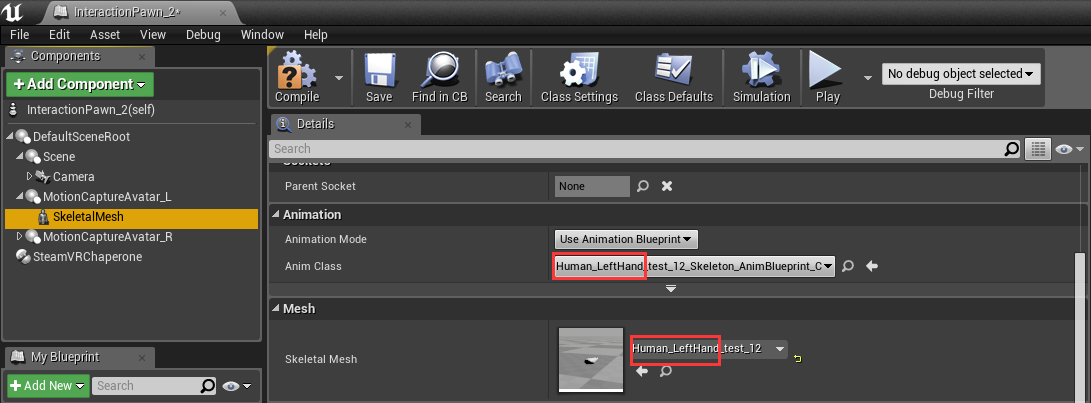
Set MotionCaptureAvatar\_L as follow: 

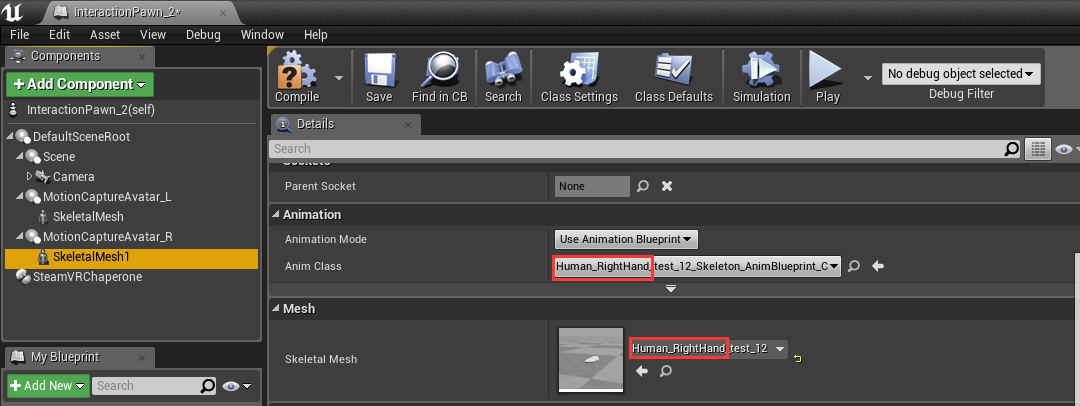
Set MotionCaptureAvatar\_R as follow:



Note: the [Avatar Name] property can be any string except NAME\_None, just need to keep the Left and Right Avatar Component has the same [Avatar Name].

Note: the [Is Local Avatar] identifies the hand pose data is from local machine or remote machine. Just leave it to true as default.

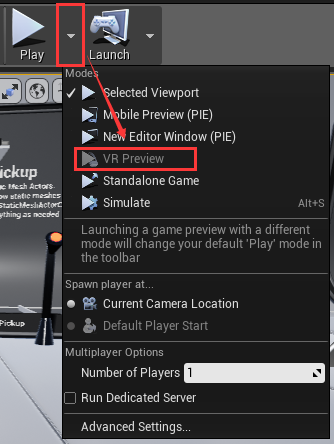
Add a **SkeletalMesh** Component to MotionCaptureAvatar\_L, then set it as follow: 

Add a **SkeletalMesh** Component to MotionCaptureAvatar\_R, then set it as follow: 

**Load Calibration Data**

After you finished the above steps, you may successfully make the hand models active, but they’re fixed at their spawn location, or to say that they have no displacement. We need calibration data which created by Hi5 sample program to solve this problem. If you have calibration data, then you just need to load it in your blueprint, then the vive tracker will drive the whole hand model realtime. **Now in MotionCapture, the calibration data will be loaded automatically. You don’t need to do it yourself.** But if you wish, you also can call [LoadCalibrationData()] api to do this work.

Compile and save project. Click [VR Preview] button on Toolbar to start game. If Hi5 dongle plugged in, we should see the Hi5 Glove drive hands model successfully.



**UE4.20 Notes**

For UE4.20, the MotionCapture plugin only can be used as Engine plugin. So please copy MotionCapture plugin to Engine plugins folder before the operations above. Eg: F:\Program Files (x86)\Epic Games\UE\_4.20\Engine\Plugins