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:
   a) The HTC VIVE system is running in a clean environment, without any optical reflection and other IR interference.
   b) The SteamVR system is running correctly.
   c) Attach the VIVE controller or VIVE tracker on each of your Hi5 glove.
   d) Turn on the attached controllers or trackers.
   e) Insert the Hi5 dongle to the USB port of your computer/ HMD.
   f) 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.

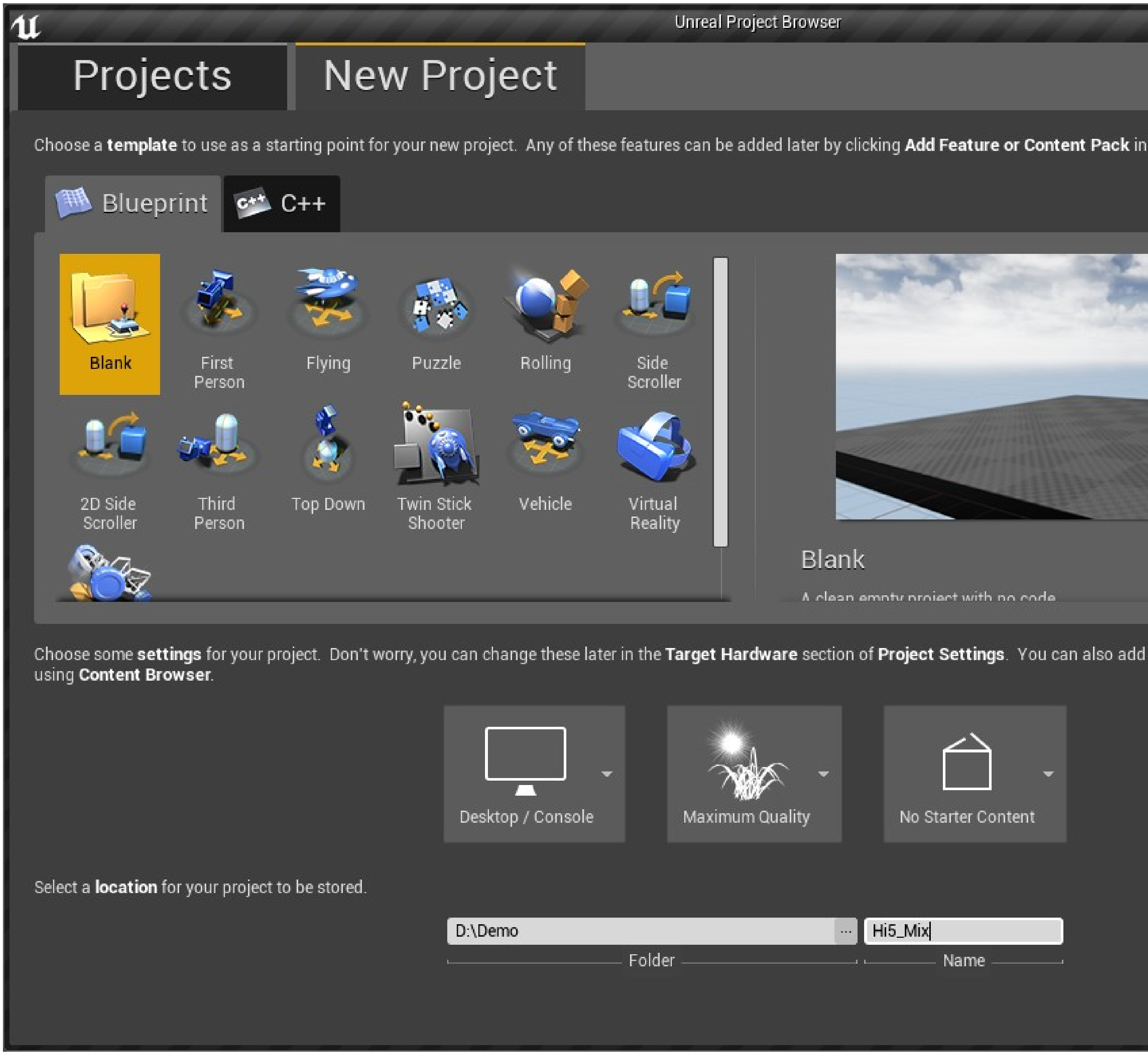
Prepare for Development

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”: ↓



Unreal Project Browser

Projects | New Project

Choose a **template** to use as a starting point for your new project. Any of these features can be added later by clicking **Add Feature or Content Pack** in

Blueprint | C++

Blank

A clean empty project with no code

Choose some **settings** for your project. Don't worry, you can change these later in the **Target Hardware** section of **Project Settings**. You can also add using **Content Browser**.

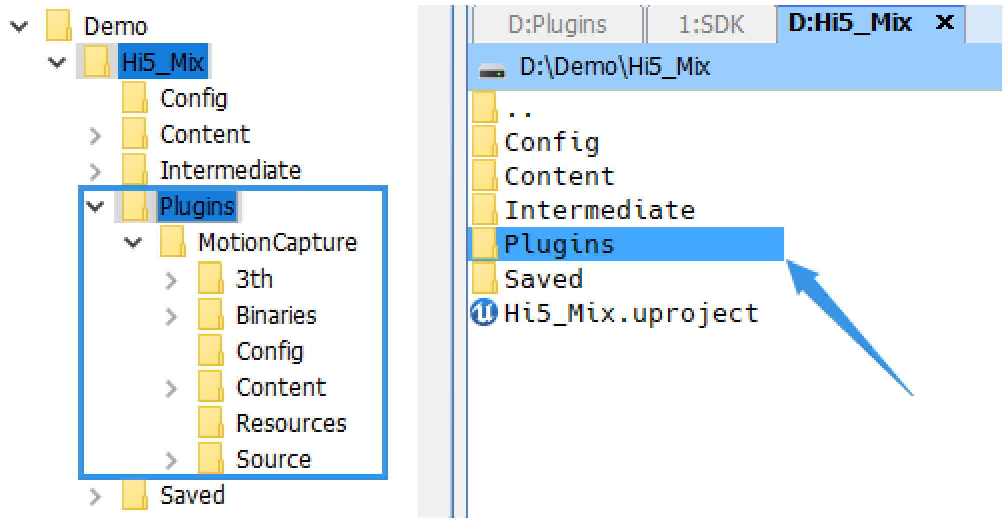
Desktop / Console | Maximum Quality | No Starter Content

Select a **location** for your project to be stored.

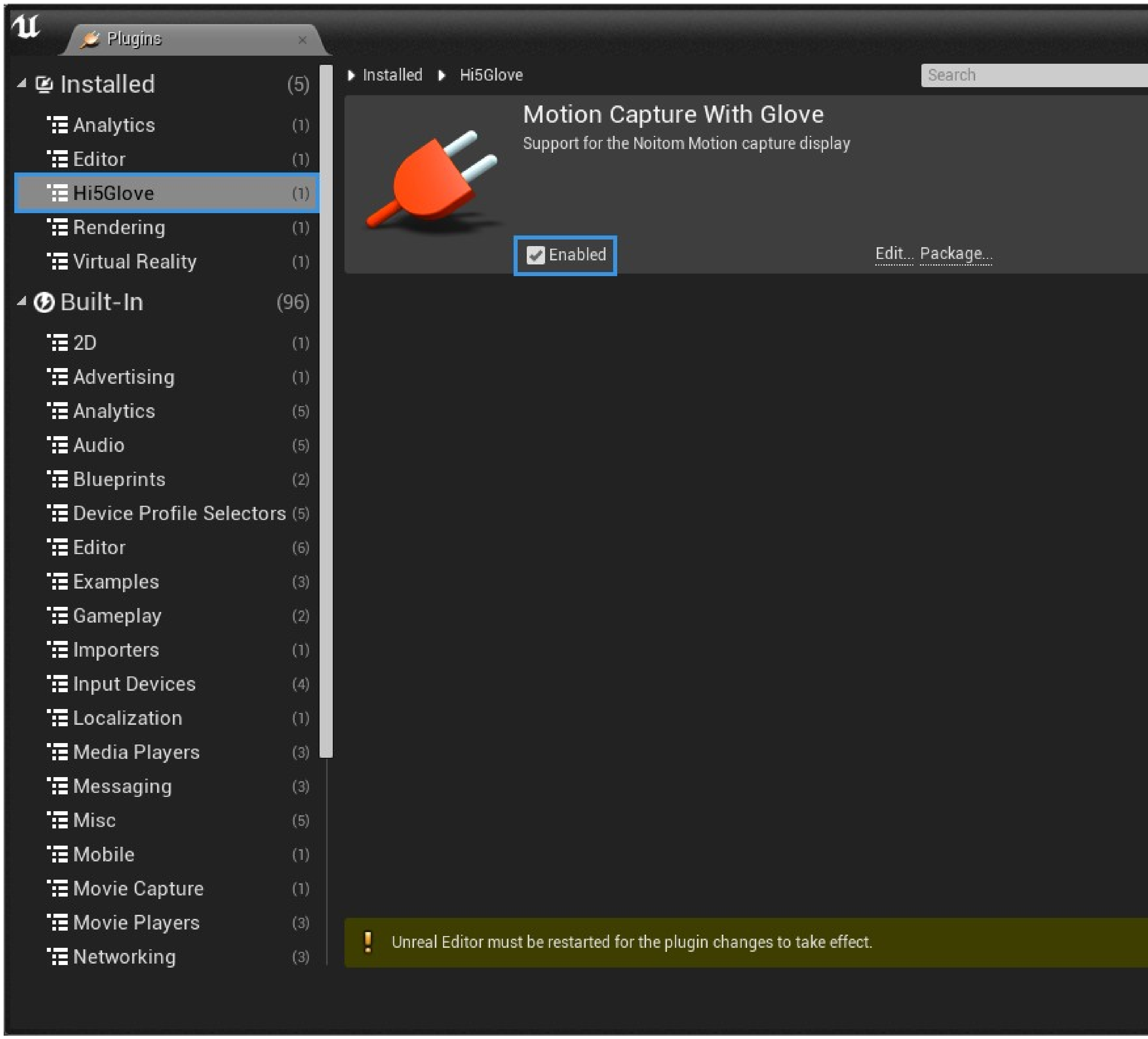
Folder: D:\Demo

Name: Hi5_Mix

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

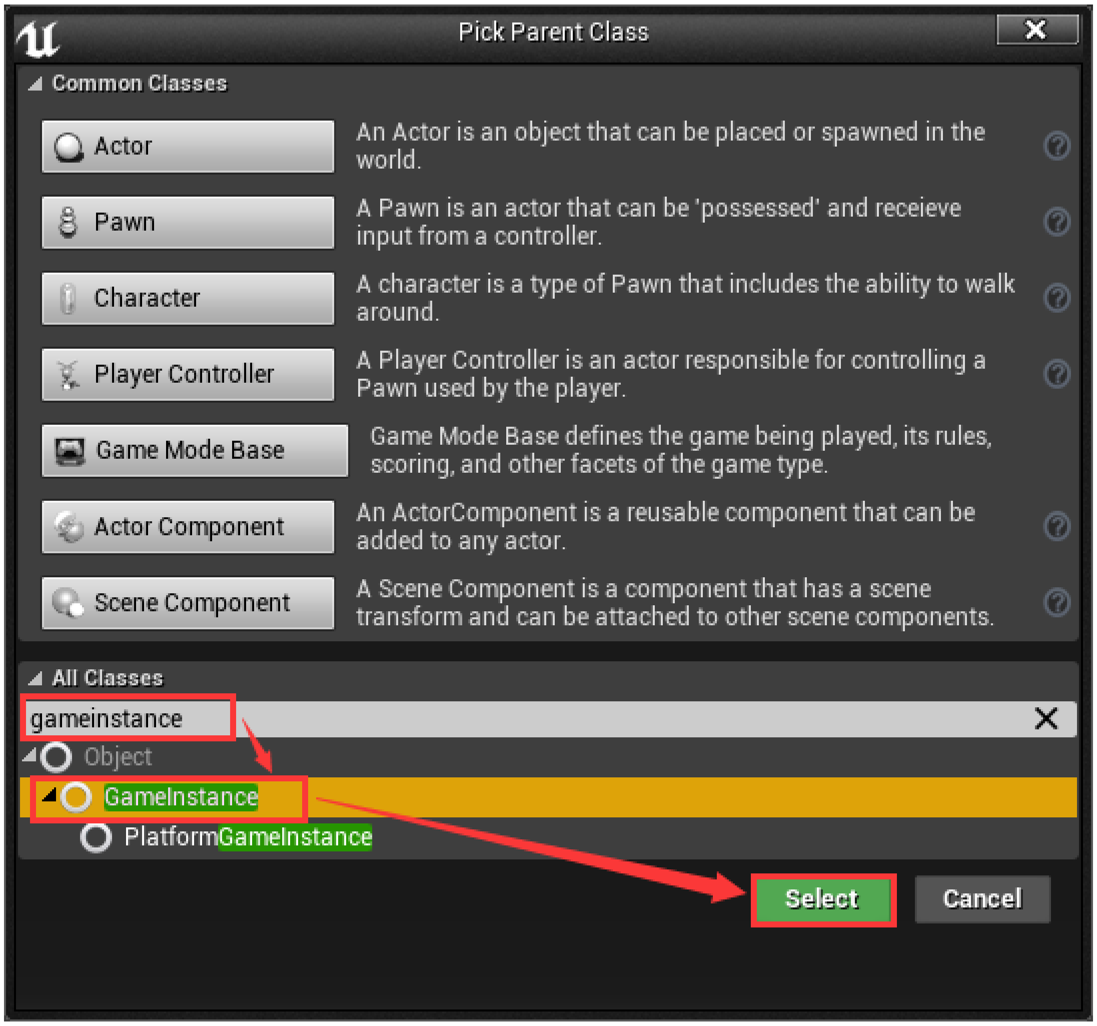


Close and Reopen the project, locate menu item: Editor -> Plugins, check if the MotionCapture plugin is listed and enabled, as follow:



Unreal Editor must be restarted for the plugin changes to take effect.

Create GameInstance



Pick Parent Class

Common Classes

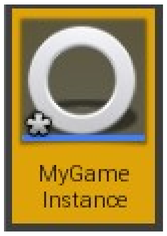
Actor	An Actor is an object that can be placed or spawned in the world.
Pawn	A Pawn is an actor that can be 'possessed' and receieve input from a controller.
Character	A character is a type of Pawn that includes the ability to walk around.
Player Controller	A Player Controller is an actor responsible for controlling a Pawn used by the player.
Game Mode Base	Game Mode Base defines the game being played, its rules, scoring, and other facets of the game type.
Actor Component	An ActorComponent is a reusable component that can be added to any actor.
Scene Component	A Scene Component is a component that has a scene transform and can be attached to other scene components.

All Classes

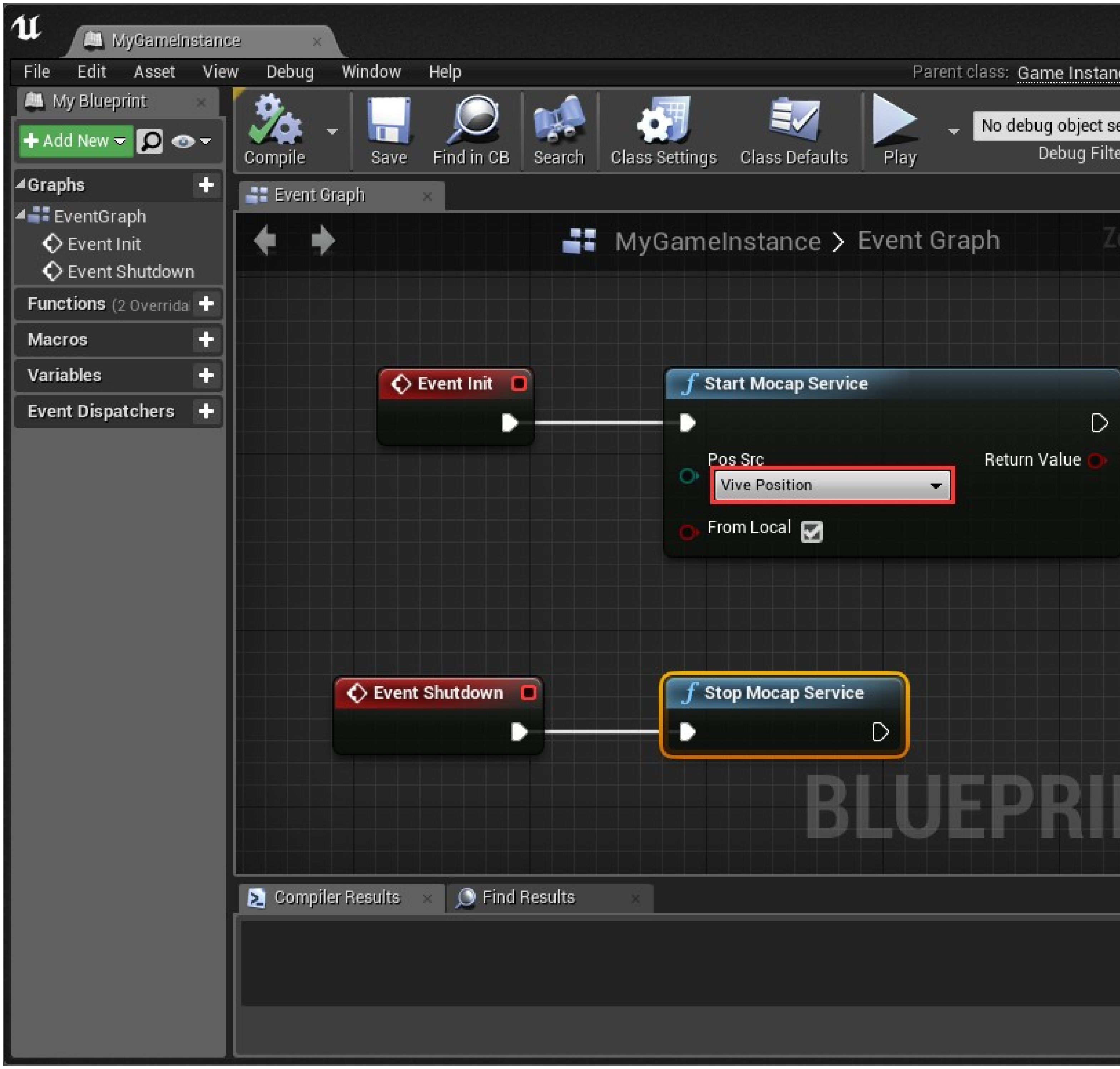
gameinstance

Object
GameInstance
PlatformGameInstance

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

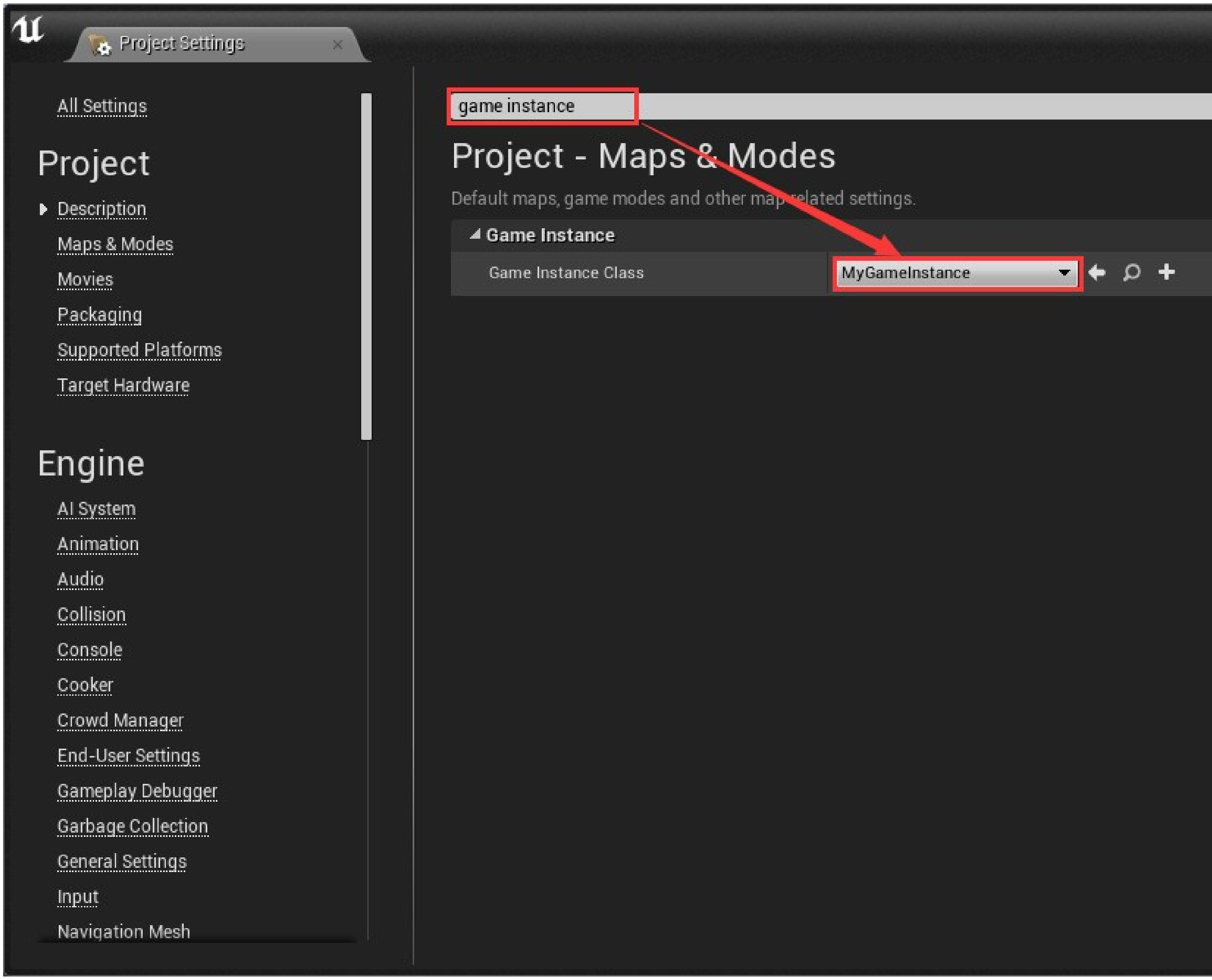


Double click the icon to edit MyGameInstace as follows:



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

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



game instance

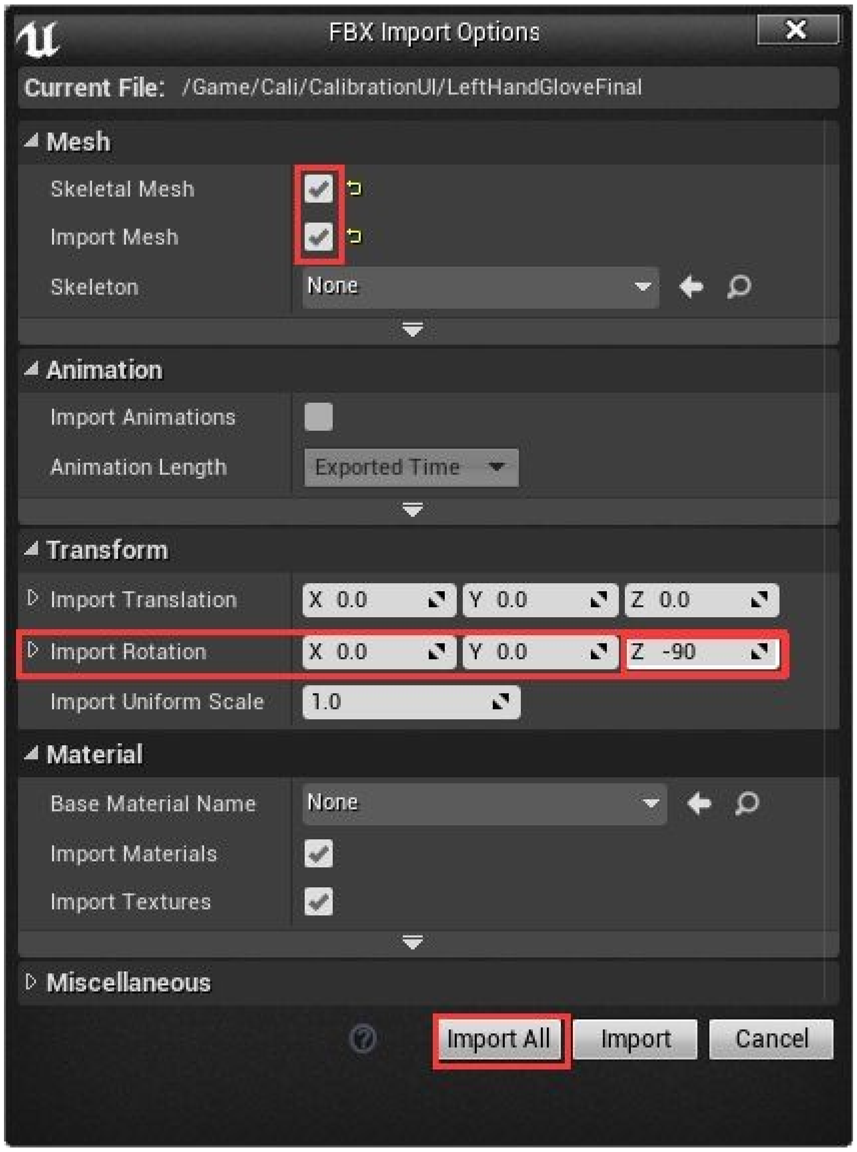
Project - Maps & Modes

Default maps, game modes and other map related settings.

Game Instance

Game Instance Class: MyGameInstance

Import Hands Models



FBX Import Options

Current File: /Game/Cali/CalibrationUI/LeftHandGloveFinal

Mesh

Skeletal Mesh	✓
Import Mesh	✓
Skeleton	None

Animation

Import Animations	
Animation Length	Exported Time

Transform

Import Translation	X 0.0	Y 0.0	Z 0.0
Import Rotation	X 0.0	Y 0.0	Z -90
Import Uniform Scale	1.0		

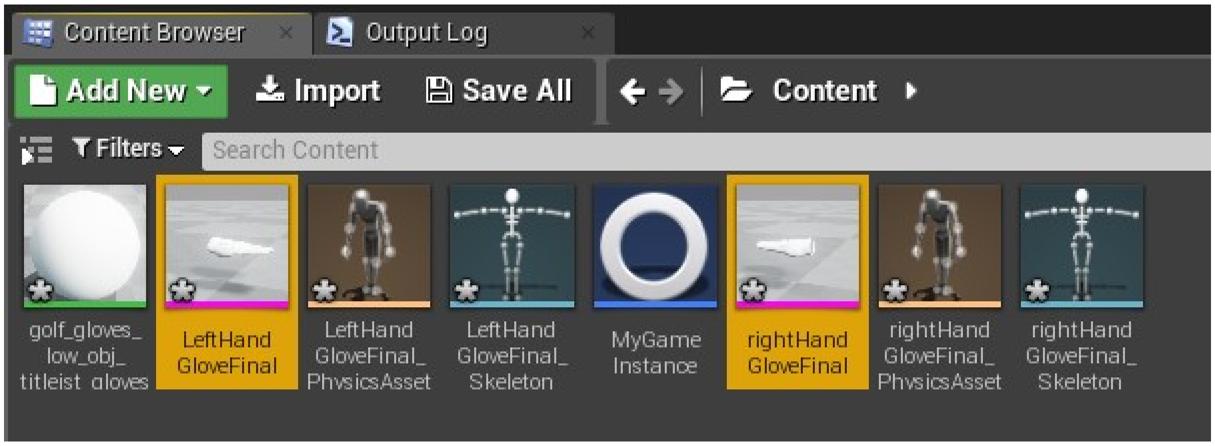
Material

Base Material Name	None
Import Materials	✓
Import Textures	✓

Miscellaneous

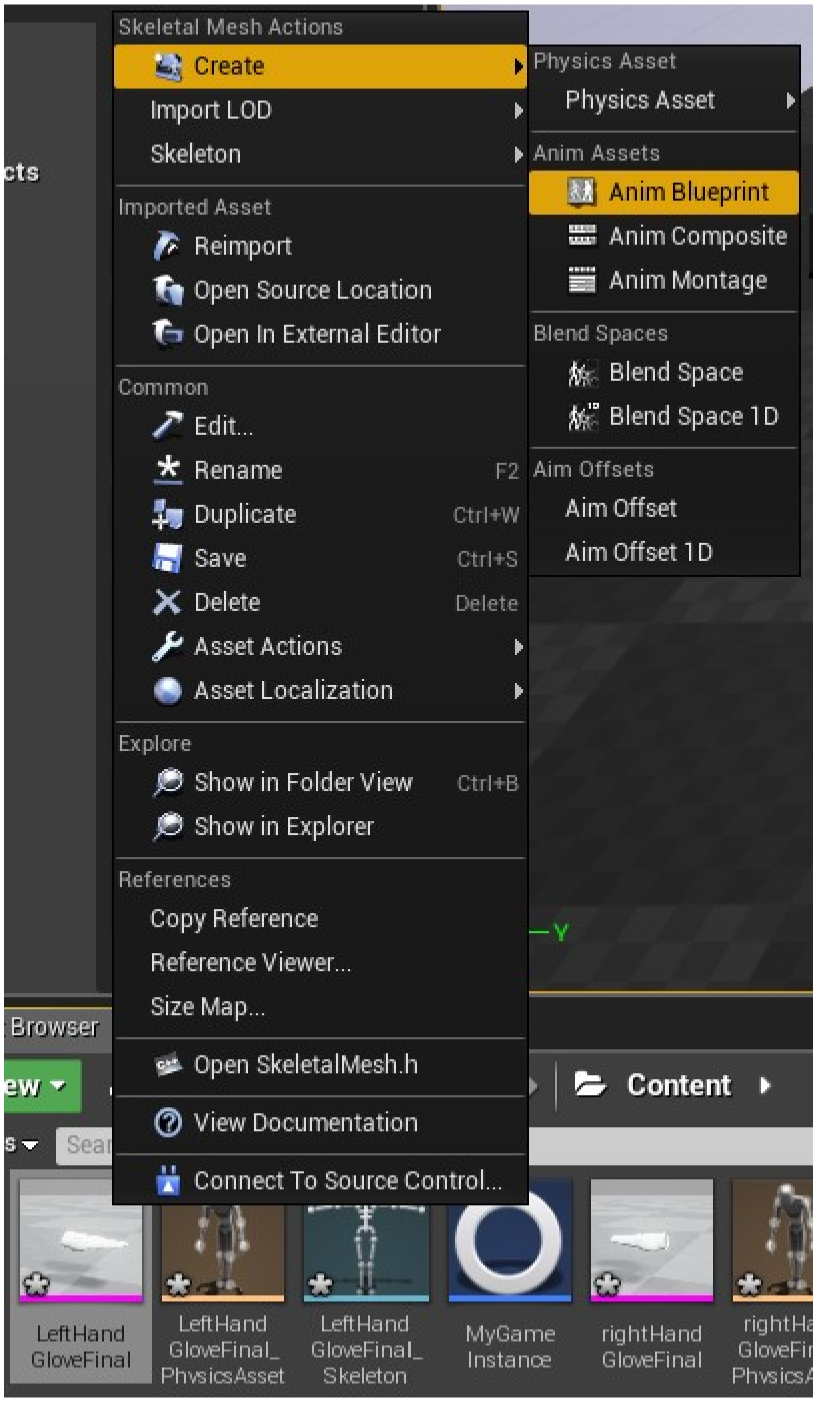
Note: the [Import Rotation] option, the Z is set to -90 degrees. This is because the imported hand models' initial orientation doesn't satisfy the requirement, So we modify the [Import Rotation] to fix the problem. The requirement of hand model's initial orientation is clear: MotionCapture plugin defines 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.)

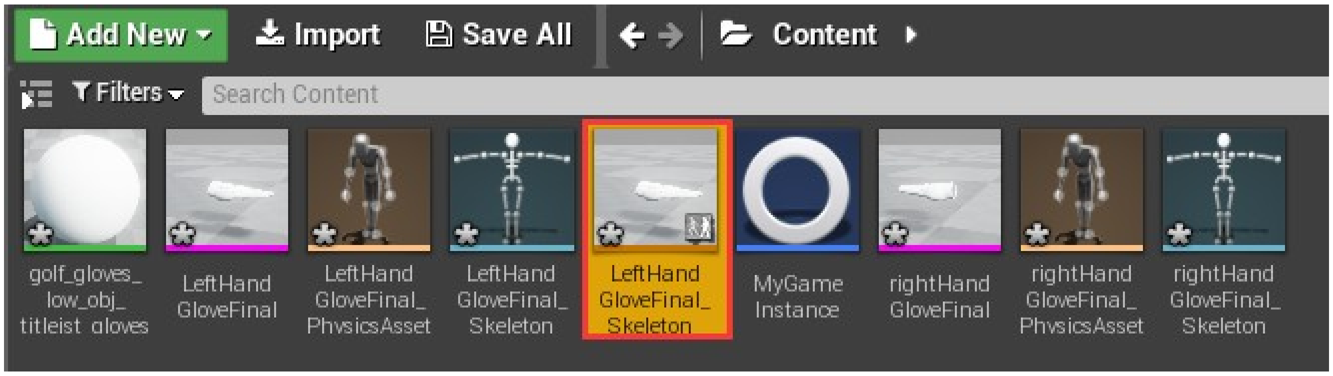

Content Browser
Output Log
Add New
Import
Save All
Content
Filters
Search Content
golf_gloves_ low_obj_ titleist_gloves
LeftHand GloveFinal
LeftHand GloveFinal_ PhysicsAsset
LeftHand GloveFinal_ Skeleton
MyGame Instance
rightHand GloveFinal
rightHand GloveFinal_ PhysicsAsset
rightHand GloveFinal_ Skeleton

Create Animation Blueprint

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



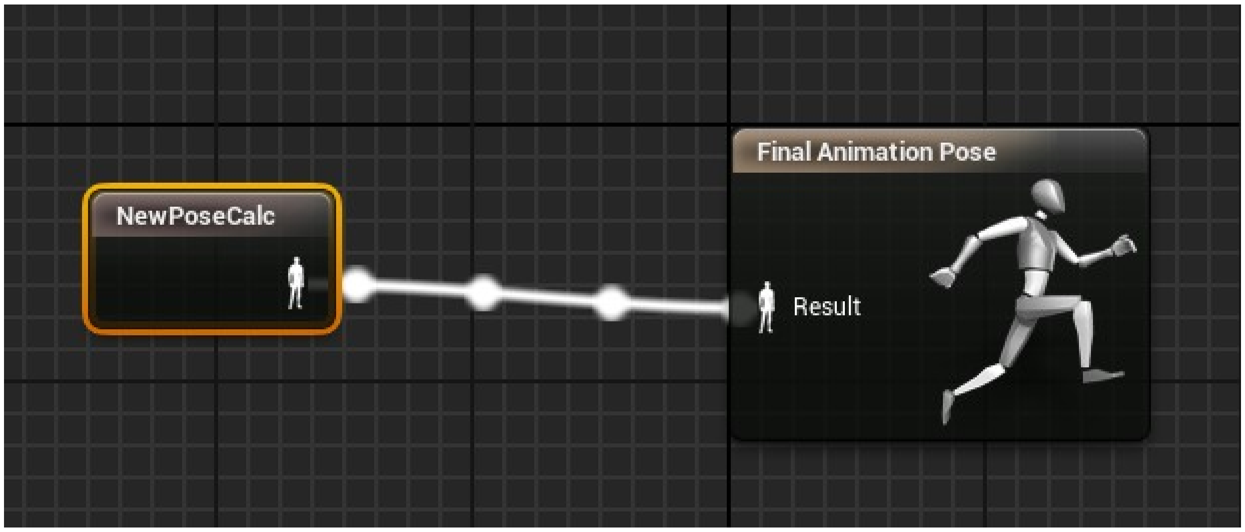
In Content Browser, we can see the newly created **LeftHandGloveFinal_Skeleton_AnimBlueprint:**



Add New
Import
Save All
Content
Filters
Search Content
golf_gloves_ low_obj_ titleist_gloves
LeftHand GloveFinal
LeftHand GloveFinal_ PhysicsAsset
LeftHand GloveFinal_ Skeleton
LeftHand GloveFinal_ Skeleton
MyGame Instance
rightHand GloveFinal
rightHand GloveFinal_ PhysicsAsset
rightHand GloveFinal_ Skeleton

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:**



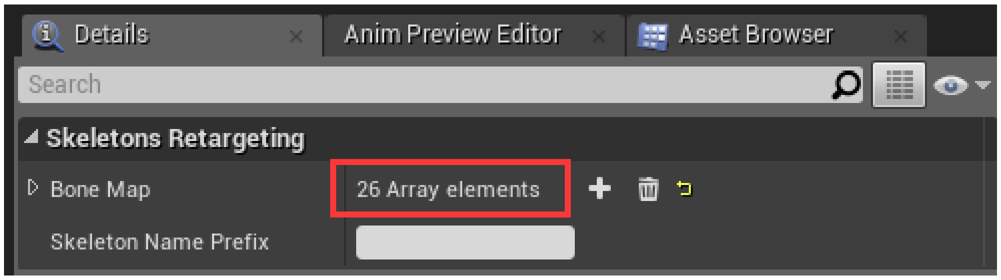
Skeleton Retargeting

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:

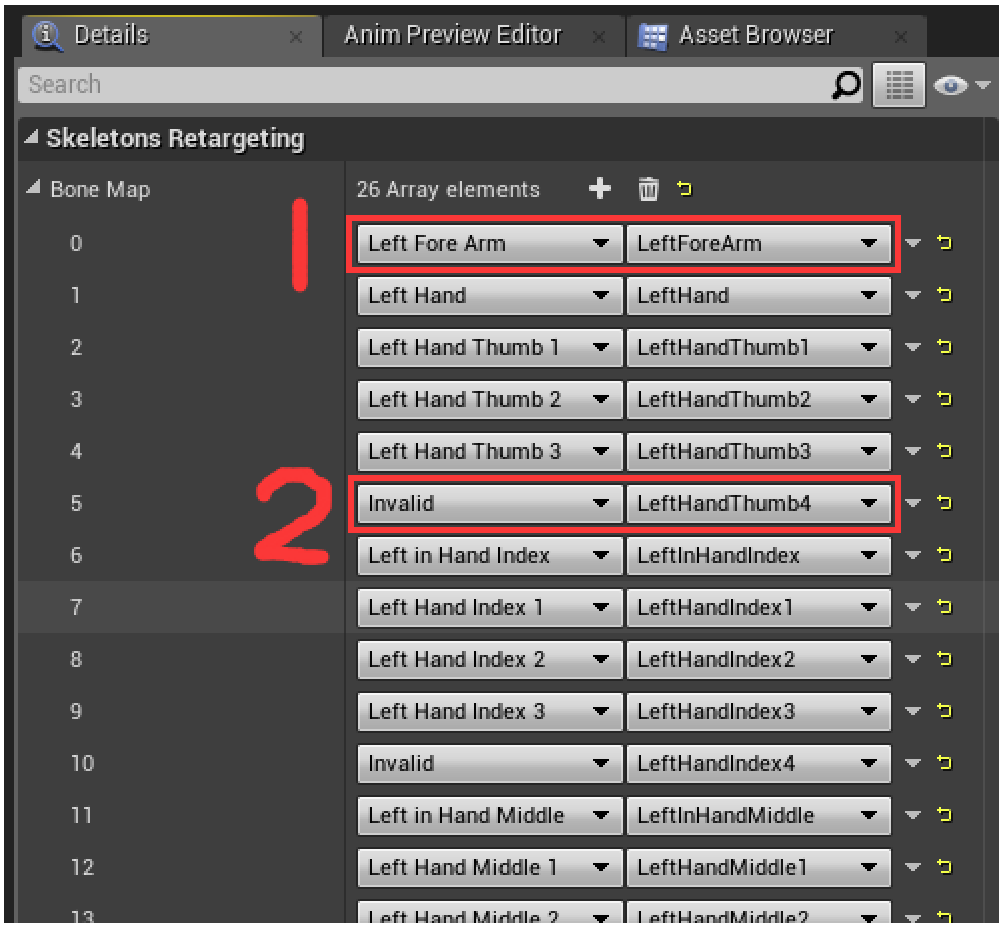


Details | Anim Preview Editor | Asset Browser

Skeletons Retargeting

Bone Map	26 Array elements
Skeleton Name Prefix	

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



Details | Anim Preview Editor | Asset Browser

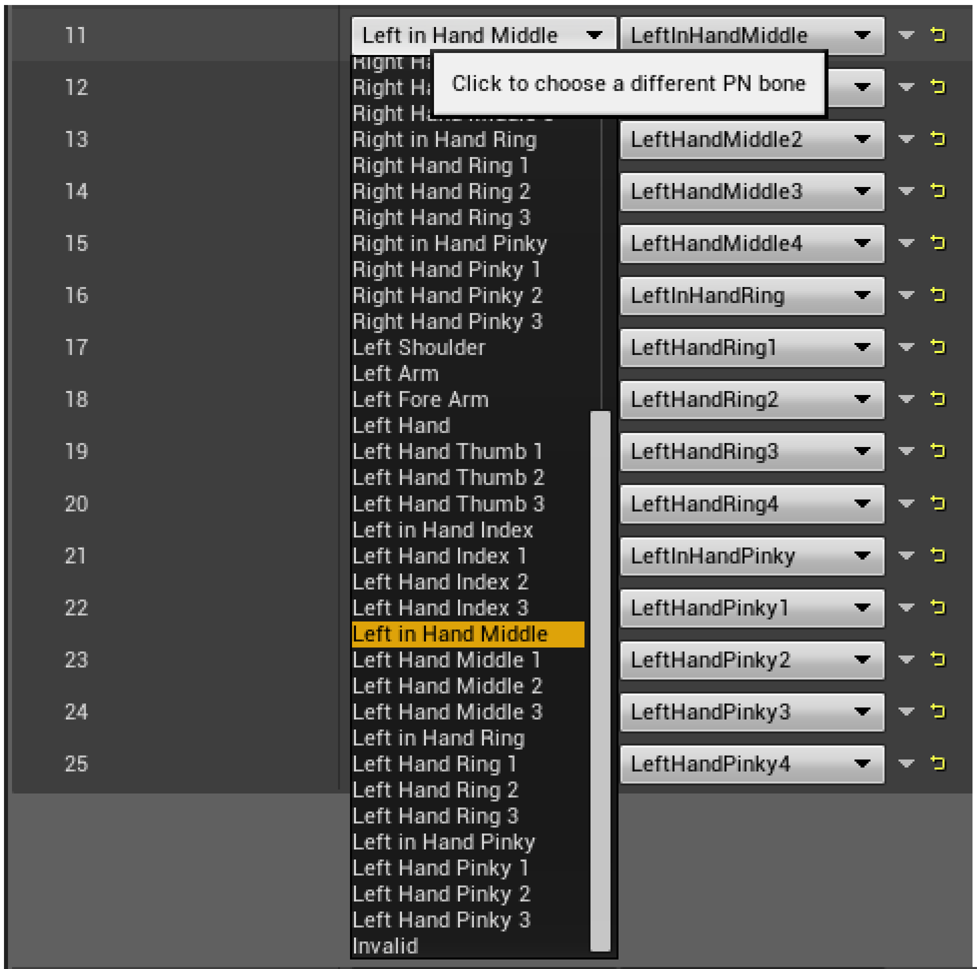
Skeletons Retargeting

Bone Map: 26 Array elements

0	Left Fore Arm	LeftForeArm
1	Left Hand	LeftHand
2	Left Hand Thumb 1	LeftHandThumb1
3	Left Hand Thumb 2	LeftHandThumb2
4	Left Hand Thumb 3	LeftHandThumb3
5	Invalid	LeftHandThumb4
6	Left in Hand Index	LeftInHandIndex
7	Left Hand Index 1	LeftHandIndex1
8	Left Hand Index 2	LeftHandIndex2
9	Left Hand Index 3	LeftHandIndex3
10	Invalid	LeftHandIndex4
11	Left in Hand Middle	LeftInHandMiddle
12	Left Hand Middle 1	LeftHandMiddle1
13	Left Hand Middle 2	LeftHandMiddle2

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.

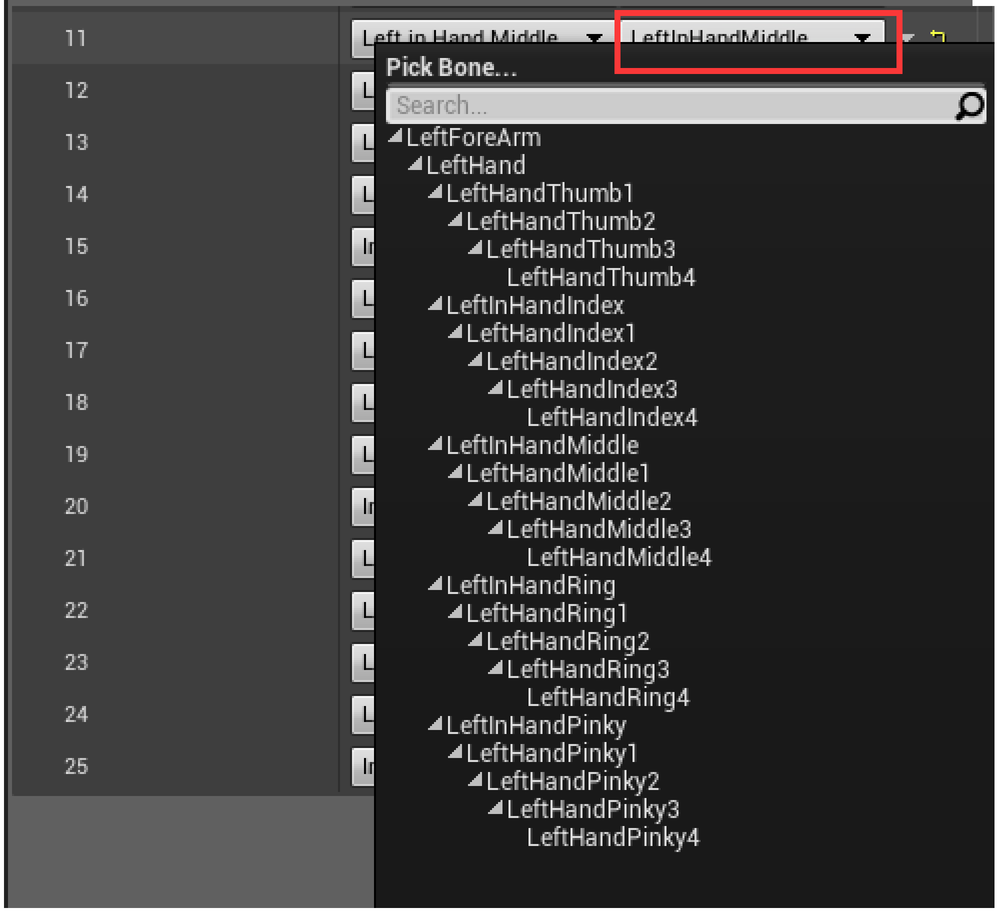
We can manually modify the corresponding relationship about Skeleton Retargeting:



11	Left in Hand Middle	LeftInHandMiddle
12		
13		LeftHandMiddle2
14		LeftHandMiddle3
15		LeftHandMiddle4
16		LeftInHandRing
17		LeftHandRing1
18		LeftHandRing2
19		LeftHandRing3
20		LeftHandRing4
21		LeftInHandPinky
22		LeftHandPinky1
23		LeftHandPinky2
24		LeftHandPinky3
25		LeftHandPinky4

Click to choose a different PN bone

Right Ha
Right Ha
Right Ha
Right in Hand Ring
Right Hand Ring 1
Right Hand Ring 2
Right Hand Ring 3
Right in Hand Pinky
Right Hand Pinky 1
Right Hand Pinky 2
Right Hand Pinky 3
Left Shoulder
Left Arm
Left Fore Arm
Left Hand
Left Hand Thumb 1
Left Hand Thumb 2
Left Hand Thumb 3
Left in Hand Index
Left Hand Index 1
Left Hand Index 2
Left Hand Index 3
Left in Hand Middle
Left Hand Middle 1
Left Hand Middle 2
Left Hand Middle 3
Left in Hand Ring
Left Hand Ring 1
Left Hand Ring 2
Left Hand Ring 3
Left in Hand Pinky
Left Hand Pinky 1
Left Hand Pinky 2
Left Hand Pinky 3
Invalid



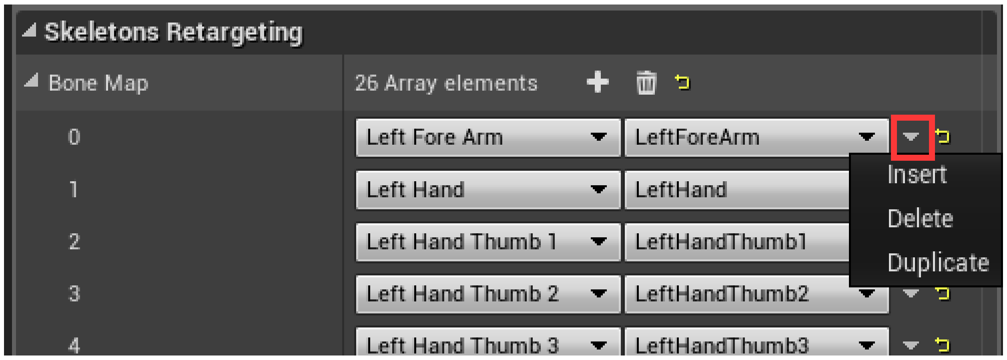
11	Left in Hand Middle	LeftInHandMiddle
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

Pick Bone...

Search...

```
◢LeftForeArm
  ◢LeftHand
    ◢LeftHandThumb1
      ◢LeftHandThumb2
        ◢LeftHandThumb3
           LeftHandThumb4
    ◢LeftInHandIndex
      ◢LeftHandIndex1
        ◢LeftHandIndex2
          ◢LeftHandIndex3
             LeftHandIndex4
    ◢LeftInHandMiddle
      ◢LeftHandMiddle1
        ◢LeftHandMiddle2
          ◢LeftHandMiddle3
             LeftHandMiddle4
    ◢LeftInHandRing
      ◢LeftHandRing1
        ◢LeftHandRing2
          ◢LeftHandRing3
             LeftHandRing4
    ◢LeftInHandPinky
      ◢LeftHandPinky1
        ◢LeftHandPinky2
          ◢LeftHandPinky3
             LeftHandPinky4
```

We also can add or delete a bone mapping:



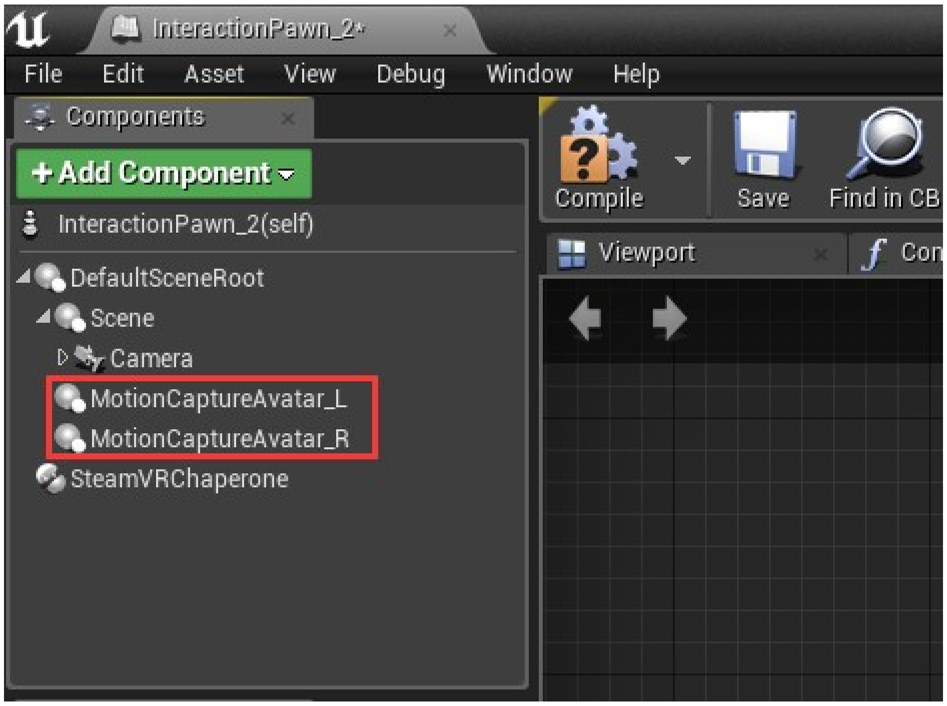
0	Left Fore Arm	LeftForeArm
1	Left Hand	LeftHand
2	Left Hand Thumb 1	LeftHandThumb1
3	Left Hand Thumb 2	LeftHandThumb2
4	Left Hand Thumb 3	LeftHandThumb3

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

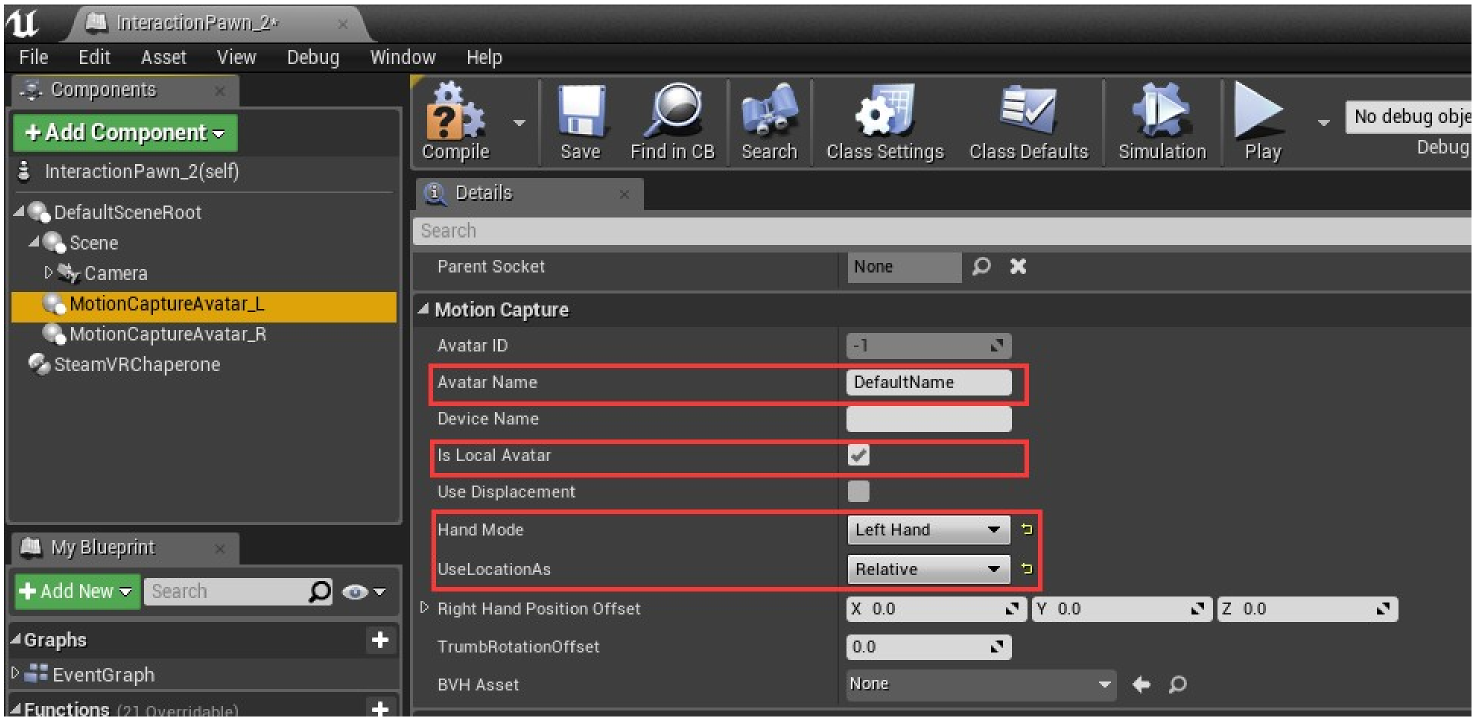
Do the same work for Right Hand.

Modify Pawn

Add two **MotionCaptureAvatar** Components to **Pawn**, then rename them to MotionCaptureAvatar_L and MotionCaptureAvatar_R.

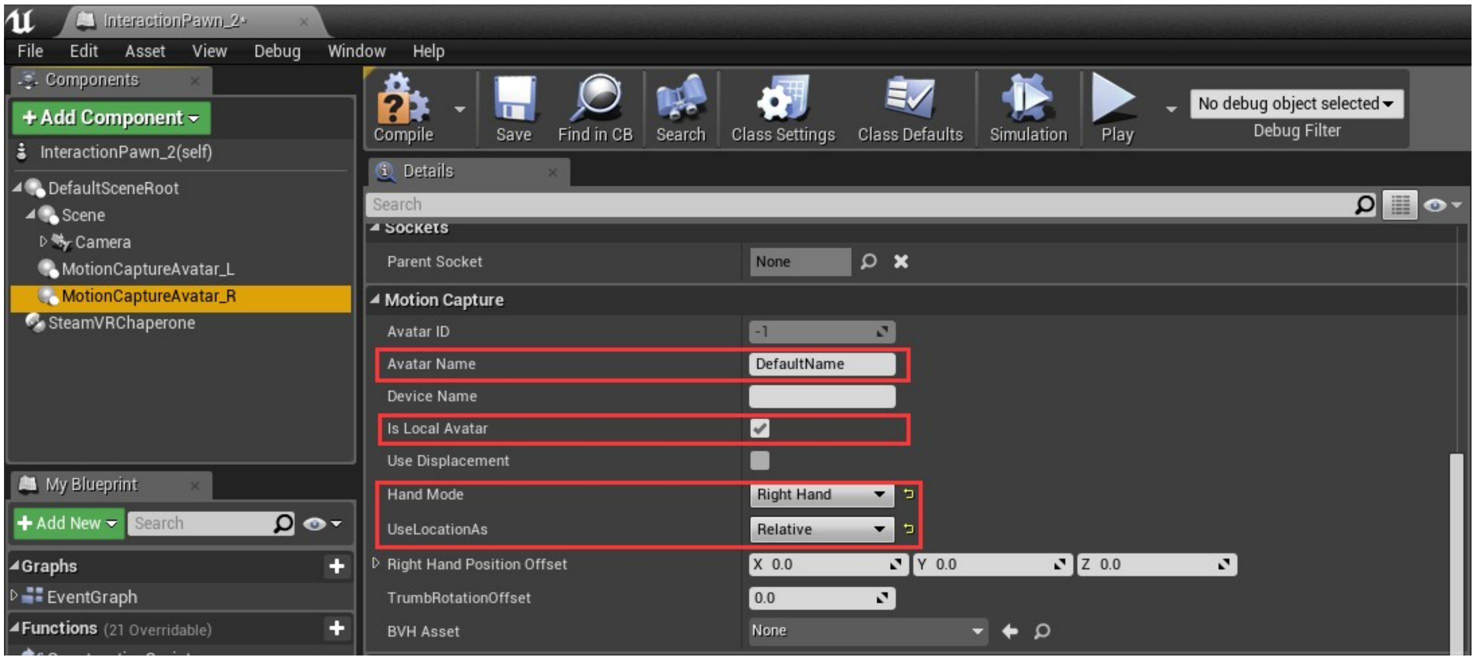


Set MotionCaptureAvatar_L as follow:



Parent Socket	None
Motion Capture	
Avatar ID	-1
Avatar Name	DefaultName
Device Name	
Is Local Avatar	✓
Use Displacement	
Hand Mode	Left Hand
UseLocationAs	Relative
Right Hand Position Offset	X 0.0 Y 0.0 Z 0.0
TrumbRotationOffset	0.0
BVH Asset	None

Set MotionCaptureAvatar_R as follow:

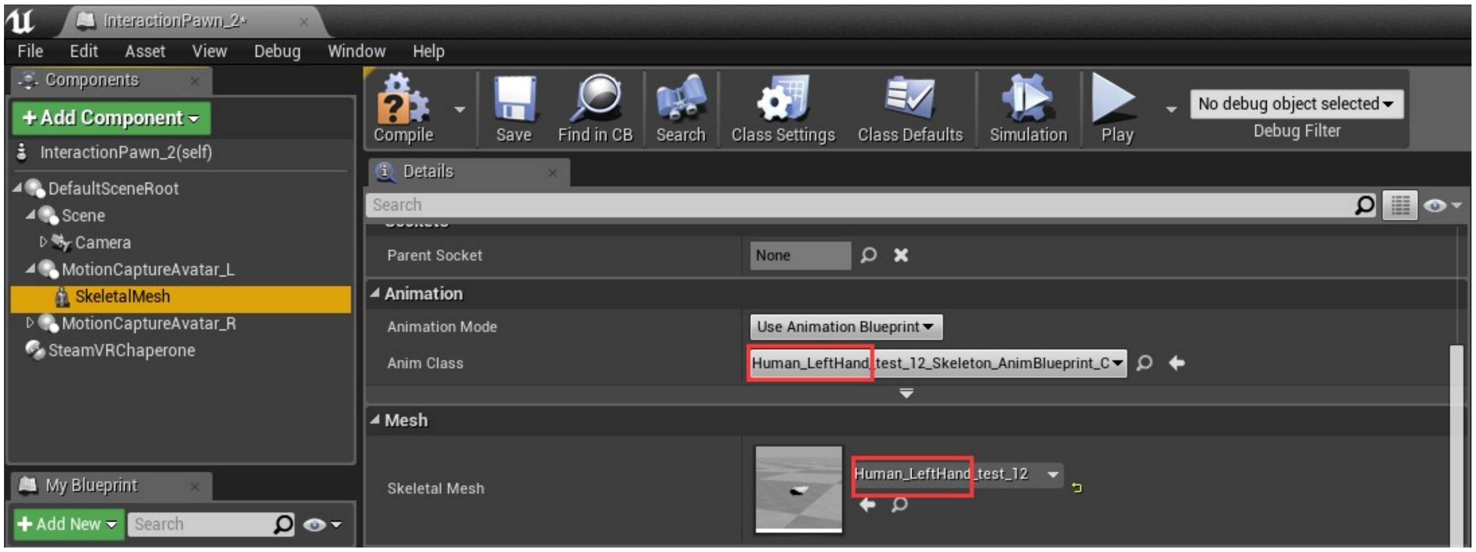


Sockets	
Parent Socket	None
Motion Capture	
Avatar ID	-1
Avatar Name	DefaultName
Device Name	
Is Local Avatar	✓
Use Displacement	
Hand Mode	Right Hand
UseLocationAs	Relative
Right Hand Position Offset	X 0.0 Y 0.0 Z 0.0
TrumbRotationOffset	0.0
BVH Asset	None

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:



Parent Socket: None

Animation

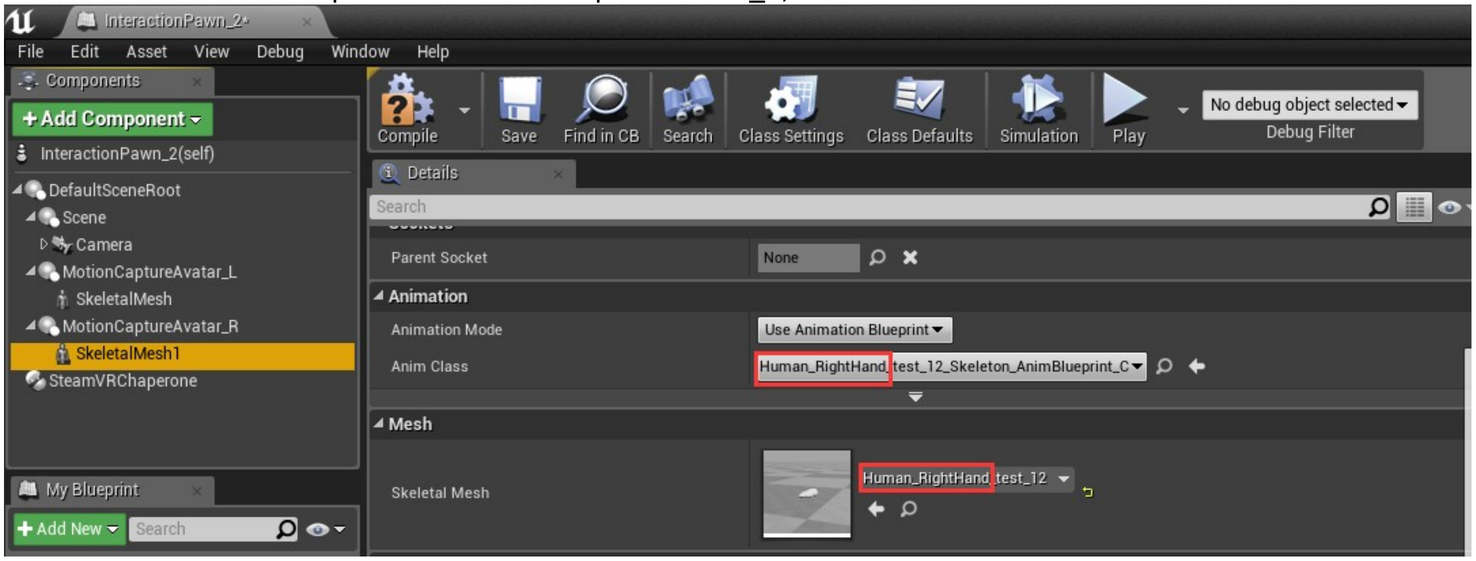
Animation Mode: Use Animation Blueprint

Anim Class: Human_LeftHand_test_12_Skeleton_AnimBlueprint_C

Mesh

Skeletal Mesh: Human_LeftHand_test_12

Add a **SkeletalMesh** Component to MotionCaptureAvatar_R, then set it as follow:



Parent Socket: None

Animation

Animation Mode: Use Animation Blueprint

Anim Class: Human_RightHand_test_12_Skeleton_AnimBlueprint_C

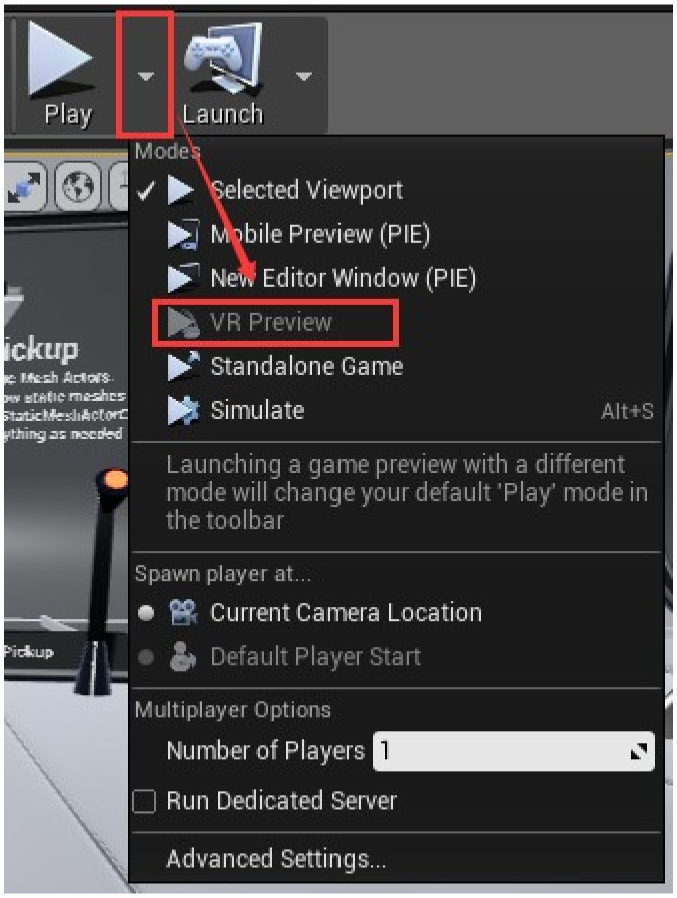
Mesh

Skeletal Mesh: Human_RightHand_test_12

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.



Launching a game preview with a different mode will change your default 'Play' mode in the toolbar

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