

HI5 2.0 INTERACTION SDK UNREAL ENGINE

STEAMVR SERIES

Contents

Overview	2
QuickStart	3
Create New Project	3
Plugin Import	3
Project Setup.....	4
Run interactive maps.....	7
Project Packaging.....	8
Interaction Prop Creation	10
Plane.....	10
Table	12
Cube	15
Sphere	19
Static Mesh.....	22
Reset Button	25
Hand model settings	28
Prop and headset settings	28
Scene switching settings.....	32
Appendix: Interactive SDK API Interface Description	34
C++.....	34
Blueprint interface.....	36

Overview

- This SDK is available for the SteamVR family of headsets, including.
 - HTC Vive Pro/ HTC Vive Pro 2/ HTC Vive Pro eye/ Pimax 8k x/ Pimax 8k plus/
 - Pimax 5k super/ Pimax ARTISAN/ Steam Valve Index
- The following items are required to use this function together:
 - Valve Base Station * 2
 - Valve Tracker/ Tundra Tracker *2
 - Head-up display*1

QuickStart

Create New Project

Create new C++ Project

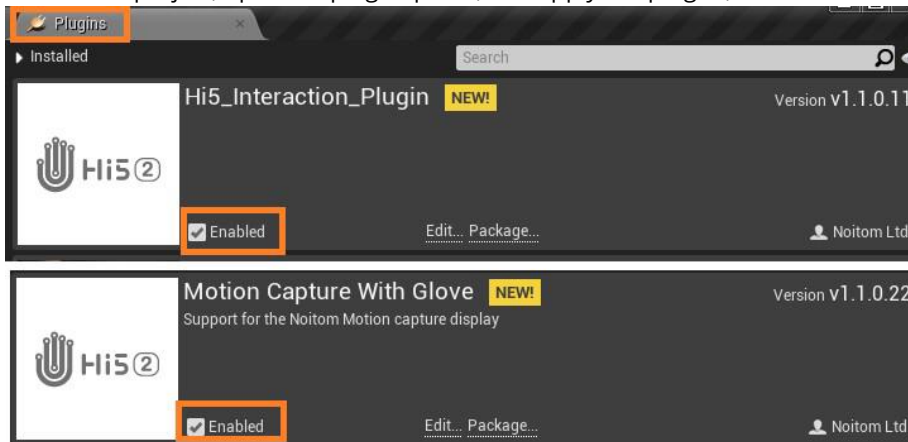
Plugin Import

1. Close the project, create a new Plugins folder in the project directory, and add the plugins Motion Capture and Hi5_Interaction_Plugin is copied to the Plugins directory, as shown in the figure:

Name	Date modified	Type	Size
.vs	1/12/2023 4:14 PM	File folder	
Binaries	1/12/2023 4:15 PM	File folder	
Config	1/12/2023 4:17 PM	File folder	
Content	1/12/2023 4:16 PM	File folder	
DerivedDataCache	1/12/2023 4:16 PM	File folder	
Intermediate	1/12/2023 4:17 PM	File folder	
Plugins	1/12/2023 4:35 PM	File folder	
Saved	1/12/2023 4:17 PM	File folder	
Source	1/12/2023 4:14 PM	File folder	
Test05_427.sln	1/12/2023 4:14 PM	Visual Studio Solu...	5 KB
Test05_427.uproject	1/12/2023 4:14 PM	Unreal Engine Proj...	1 KB

Name	Date modified	Type	Size
Hi5_Interaction_Plugin	1/12/2023 4:35 PM	File folder	
MotionCapture	1/12/2023 4:34 PM	File folder	

2. Restart the project, open the plug-in panel, and apply the plug-in, as shown in the figure:



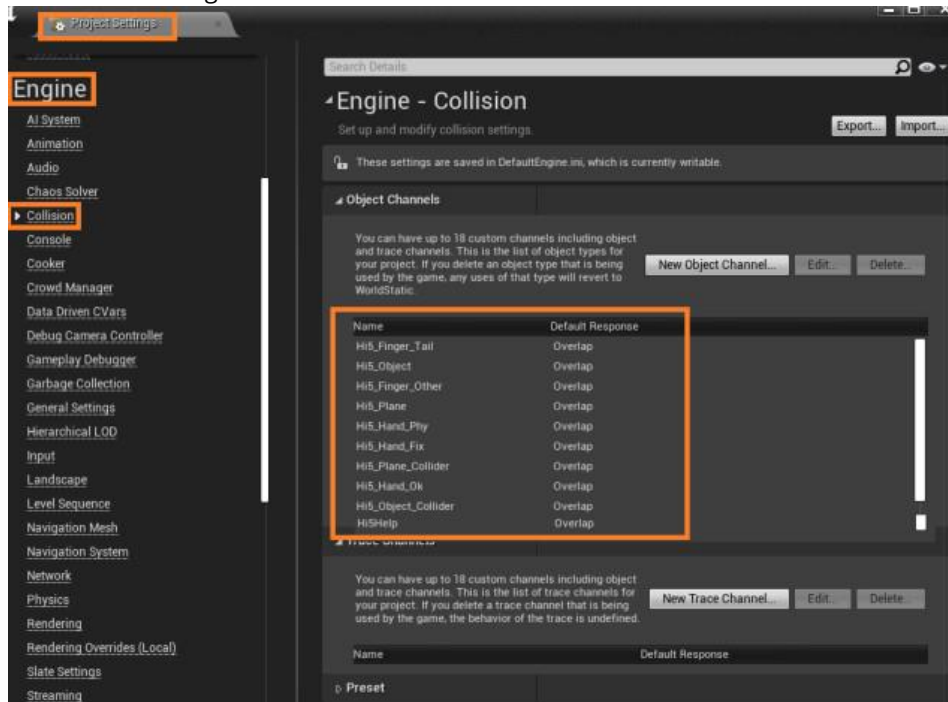
3. Apply the SteamVR plugging shown here:



Project Setup

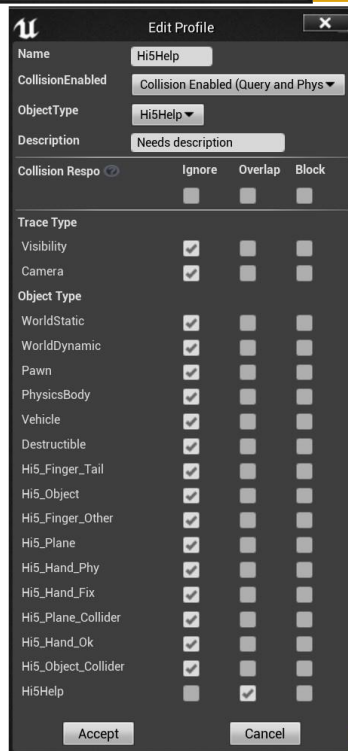
1. Engine collision settings: Engine collision can be set in two ways:
 - 1) By replacing the engine configuration file (recommended): replace the DefaultEngine.ini file in the Interaction SDK with the file of the same name in the Config folder of the UE project, and restart the project after the replacement is completed, as shown in the figure:
 - 2) If there are other engine collision settings, you cannot use method 1 to directly replace the configuration file, you need to manually complete the configuration in the Engine->collision panel, the specific operation is to click the NewObjectChannel button to add the following ten Object
 - 3) Channels in turn, Default Response are set to Overlap:
 - i. Hi5_Finger_Tail
 - ii. Hi5_Object
 - iii. Hi5_Finger_Other
 - iv. Hi5_Plane
 - v. Hi5_Hand_Phy
 - vi. Hi5_Hand_Fix
 - vii. Hi5_Plane_Collider
 - viii. Hi5_Hand_Ok
 - ix. Hi5_Object_Collider
 - x. Hi5Help

As shown in the figure:



2. After Object Channels are added, you need to complete each Preset setting in turn, double-click to open the Object Channel just created, and then refer to the following figure to complete the parameter editing, as shown in the figure:

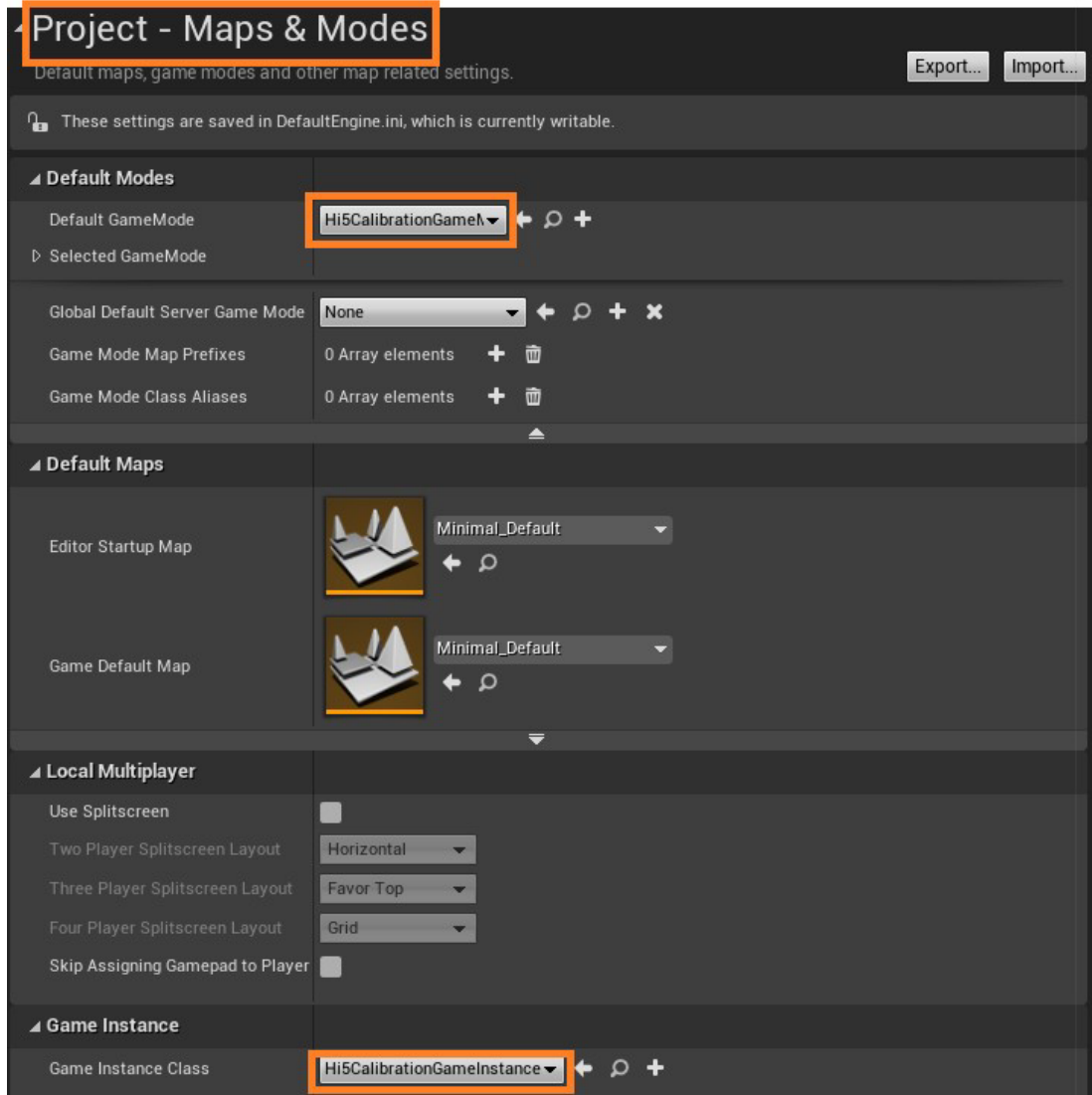




3. Maps&Mode Settings:

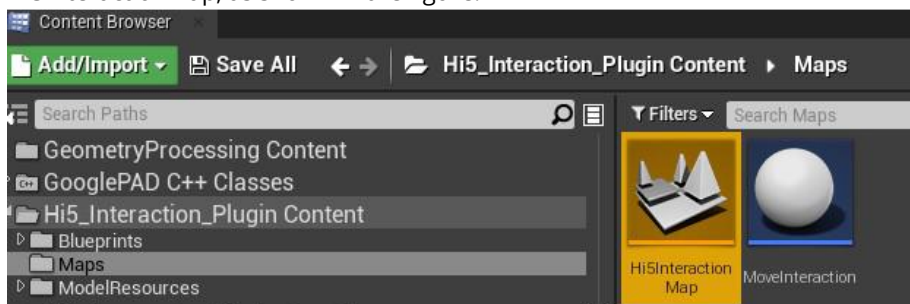
- 1) Default GameMode is set to Hi5CalibrationGameMode
- 2) Game Instance Class is set to Hi5CalibrationGameInstance

As shown in the figure:

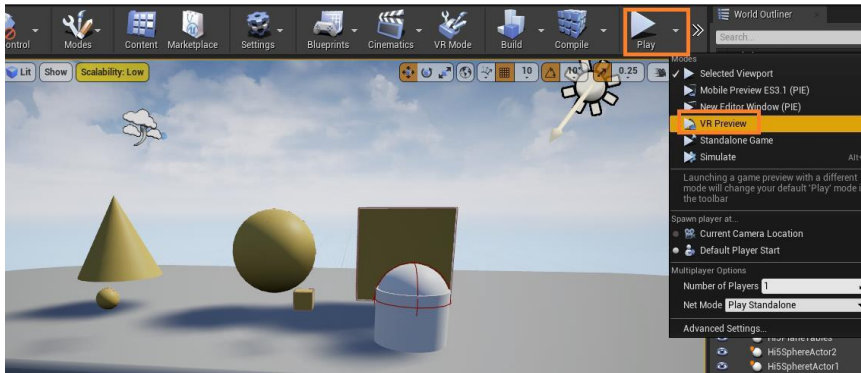


Run interactive maps

1. Content Browser ->Hi5_Interaction_Plugin Content ->Maps in open interaction map Hi5InteractionMap, as shown in the figure:



2. Click Play->VR Preview to run, as shown in the figure:

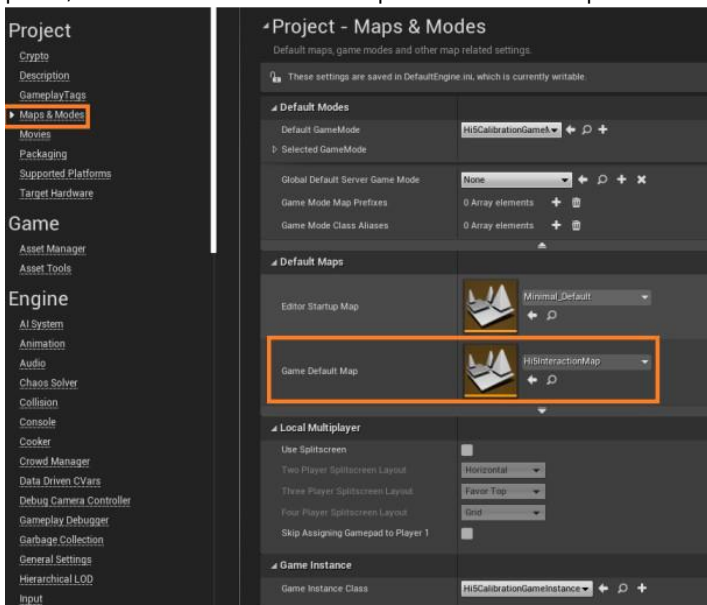


**Note: If the VR mode gray does not click, please check if the headset connection cable is properly connected and if the SteamVR software is working properly*

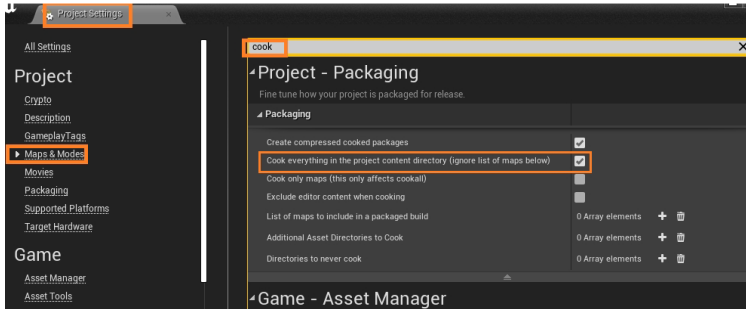
3. Keyboard 1 key to switch to the calibration scene, after finishing the calibration, keyboard 2 key to switch to the interactive use scene, and then you can experience Hi5_2 UnrealEngine interactive SDK functions in VR Preview mode

Project Packaging

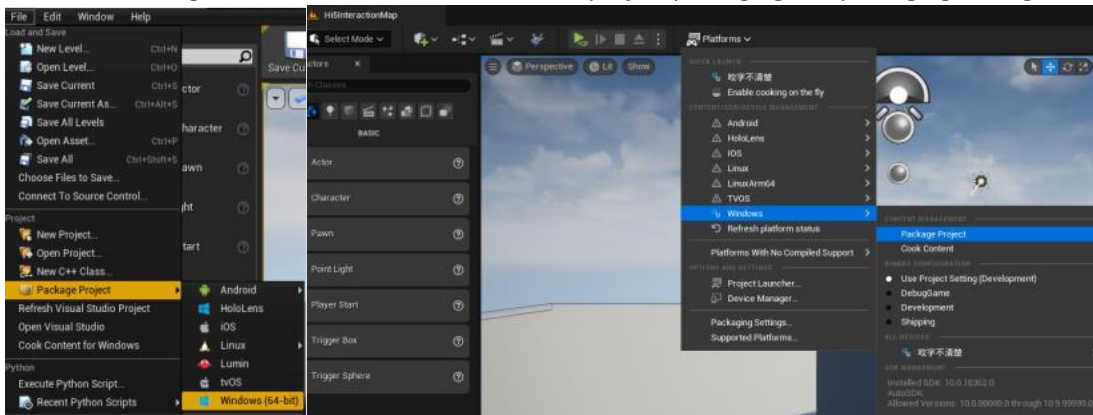
1. Open the project settings interface, set the Game Default Map in the Map&Modes panel, either as Hi5InteractionMap or CalibrationMap



- Search for the keyword cook in the Map&Modes panel, then find the checkbox Cook everything in the project content directory (ignore list of maps below), as shown in the figure:



- Click File->Package->Windows(64-bit) to execute the project packaging, the packaging settings are as

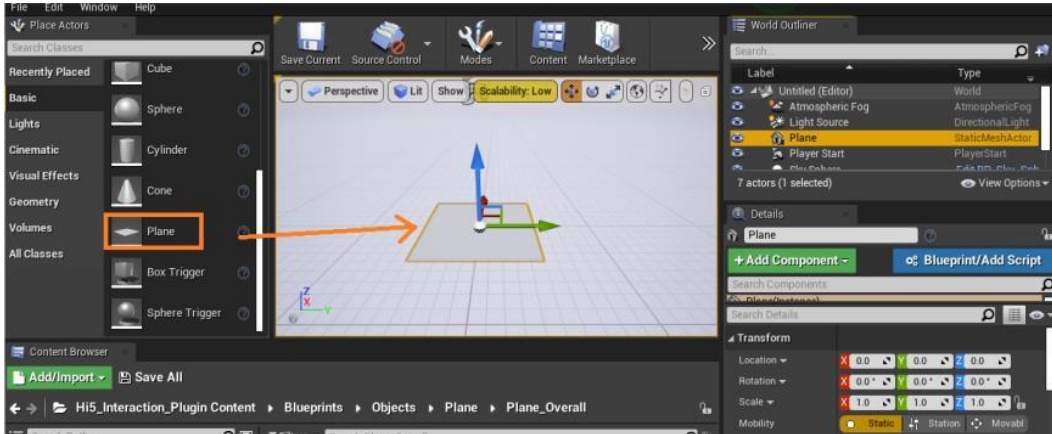


- shown in the figure:
 *Note: The figure on the left shows the UE4 packing setup, and the figure on the right shows the UE5 packing setup

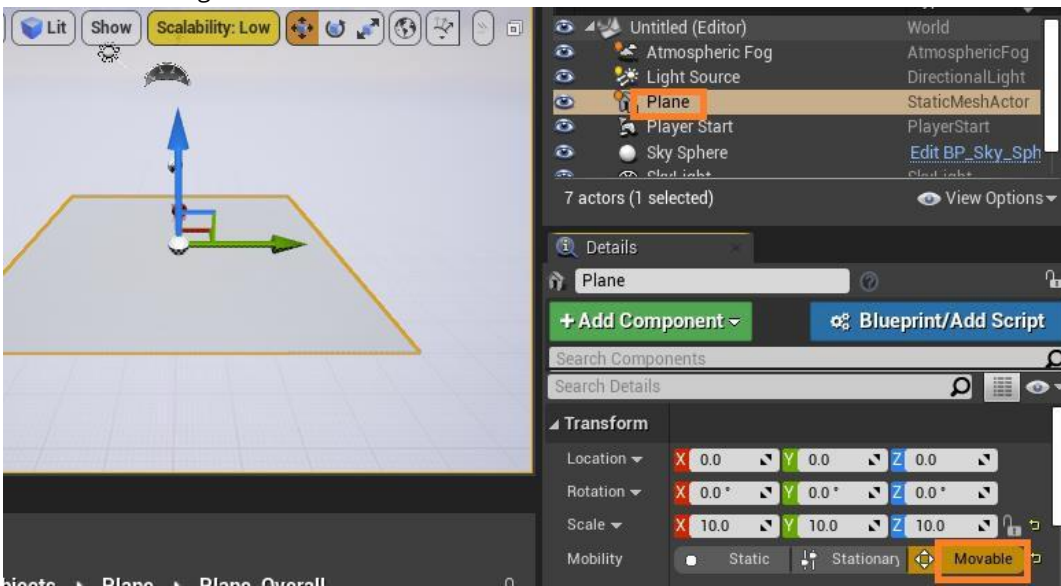
Interaction Prop Creation

Plane

1. Drag the **Plane** in the **Place Actor** panel and place it in the viewport, as shown in the figure

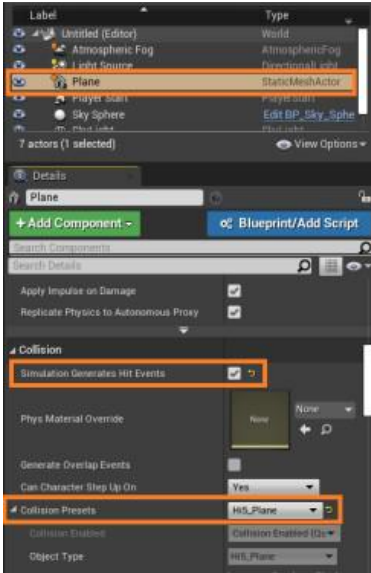


2. Adjust the floor to the appropriate size in the Details --> Transform panel, and set Mobility to Movable, as shown in the figure Shown:

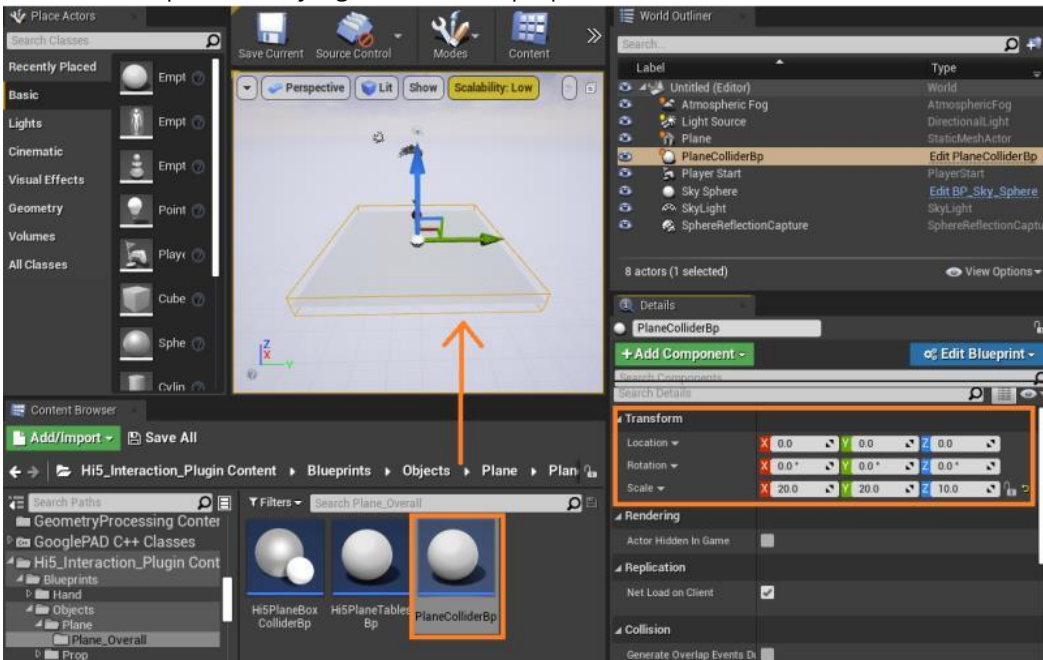


**Note: The purpose of setting the floor to be movable is to better adapt to the scene prop reset function in the interactive SDK, that is, the function of the Reset button on the back of the left hand.*

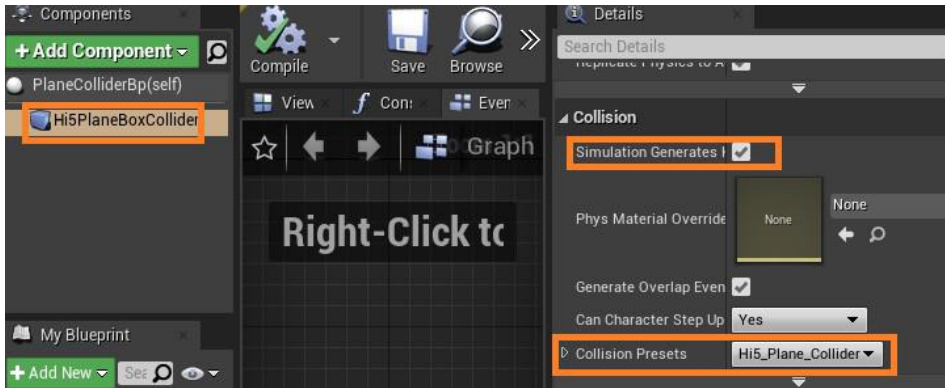
3. Complete the collision relationship setting in the Details>Collision panel, enable **Simulation Generates Hit Events** after checking other props Actors collide with the ground, events can occur. Collision Presets are set to Hi5_Plane as shown in the figure:



- Go to **Hi5_Interaction_Plugin_Content->Blueprints->Objects->Plane->Plane_Overall** directory. Drag PlaneColliderBp into the viewport and adjust the positional relationship with Plane. PlaneColliderBp is used to judge whether the prop is in contact with the floor, as shown in the figure:

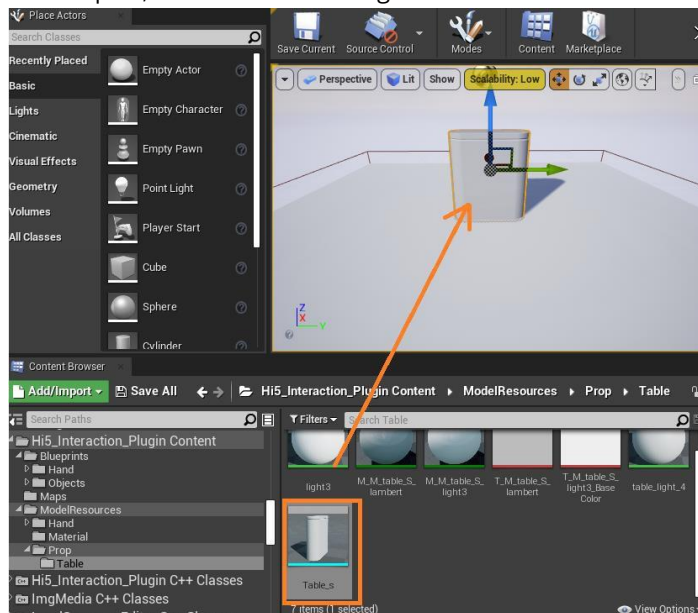


- Open PlaneColliderBp, select Hi5PlaneBoxCollider, check Simulation Generates Hit Events Collision Presets in Details-->Collision and set it to Hi5_Plane_Collider, as shown in the figure:

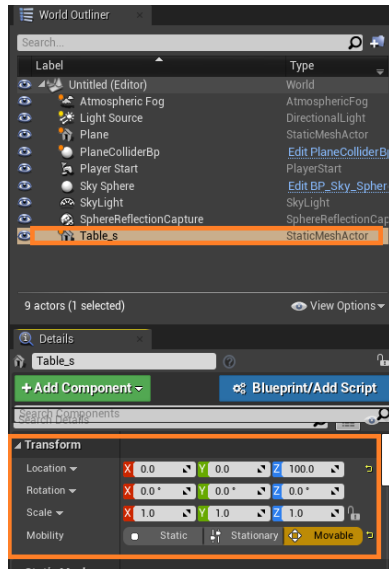


Table

1. Hi5_Interaction_Plugin content->ModeResource->Prop->Table directory, drag the Table_s model into the viewport, as shown in the image:

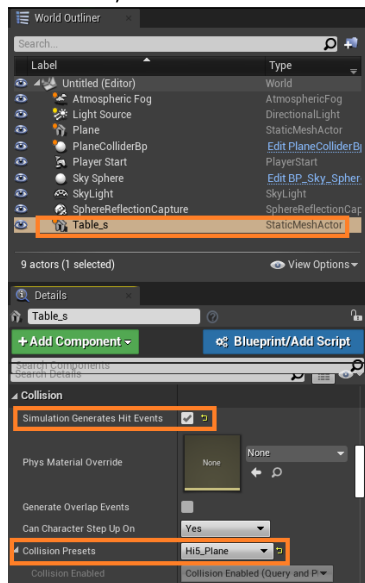


2. Check Table_S in World Outliner and complete the following settings in the Details panel:
 - 1) In Transform, adjust the floor to the right size, and Mobility set to Movable, as shown in the image:



*Note: The purpose of setting the table movable is to better adapt to the scene prop reset function in the interactive SDK, that is, the Reset button function on the back of the left hand.

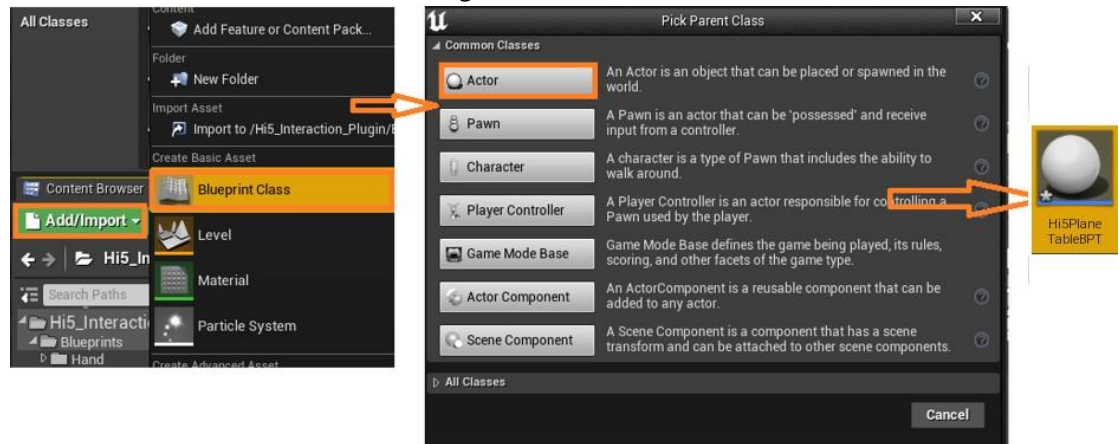
- 2) Details-> Collision panel to complete the collision relationship setting, check Simulation Generates Hit Events, after checking, when other prop Actors collide with the ground, events can occur, and Collision Presets is set to Hi5_Plane, as shown in the figure:



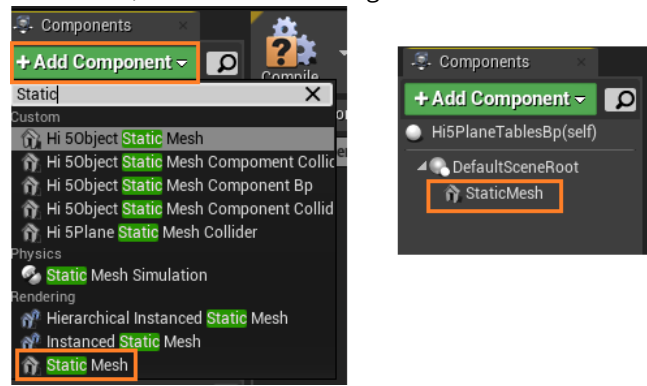
3. In this example, we will take the Table_S preset in the interactive SDK as an example to create an Actor Blueprint class to determine whether the object is in contact with the desktop, and the specific operation is as follows:

- 1) Click the Add button in the Content Browser to create an Actor Blueprint class, named in this example

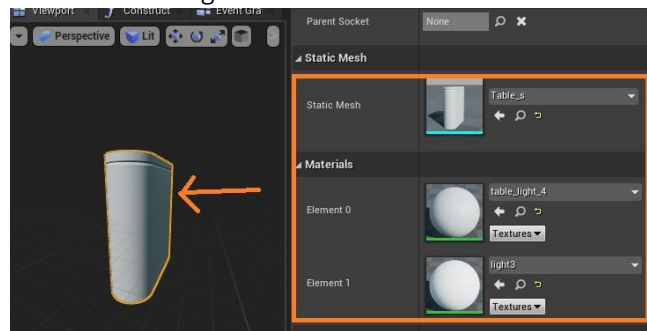
Hi5PlaneTablesBPT, as shown in the image:



- 2) Double-click to open the Actor Blueprint Hi5PlaneTablesBPT, click Add Component to add StaticMesh, as shown in the image:

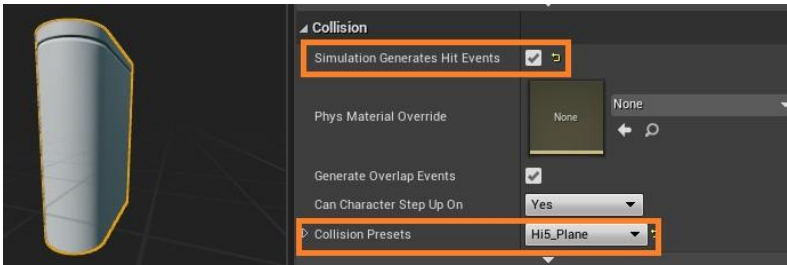


- 3) Select the StaticMesh you just added to apply the prepared table material to the Actor, as shown in the image:



4. Check Simulation Generates Hit Events in the Collision panel, and Collision Presets is set to

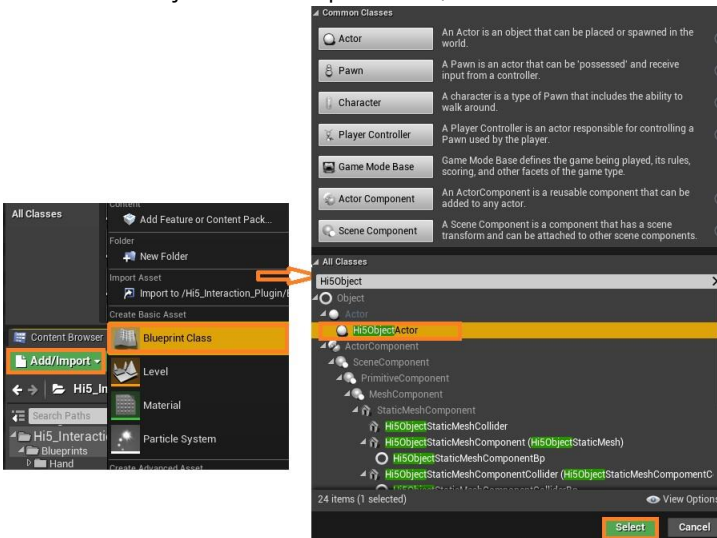
Hi5_Plane, as shown in the image:



5. Compile and save
6. Drag the created Hi5PlaneTablesBPT into the viewport to coincide with Table_s, as shown in the figure:

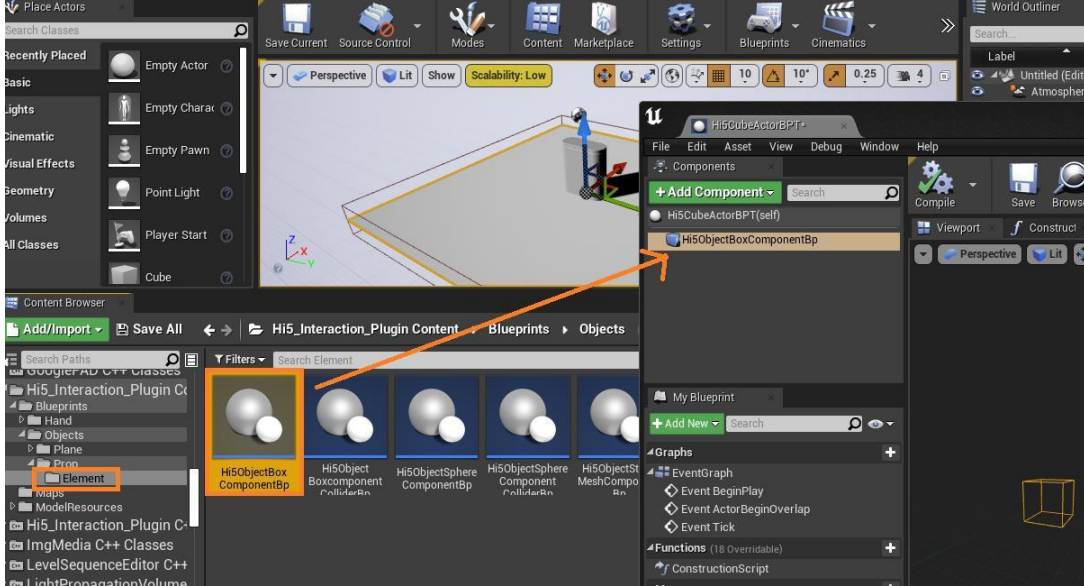
Cube

1. Create a Hi5ObjectActor Blueprint class, named Hi5CubeActorBPT here, as shown in the image:



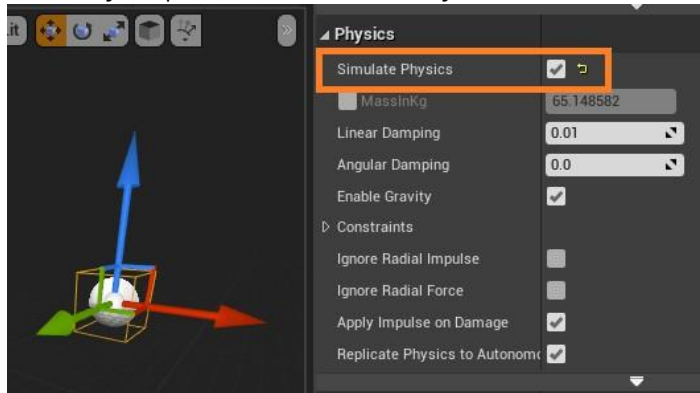
2. Double Hit Open just created blue image class Hi5CubeActorBPT, which will interact with the prebuilt SDK

Hi5ObjectBoxComponentBp as drag and drop to the Hi5CubeActorBp root directory, as shown in the figure:

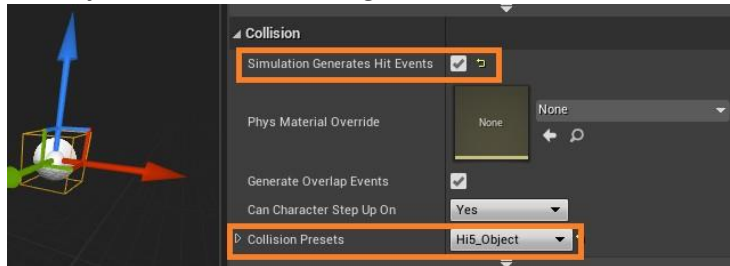


3. Hi5CubeActBPT Select Hi5ObjectBoxComponentBp in the Blueprint class, and complete the following settings in the Details panel on the right:

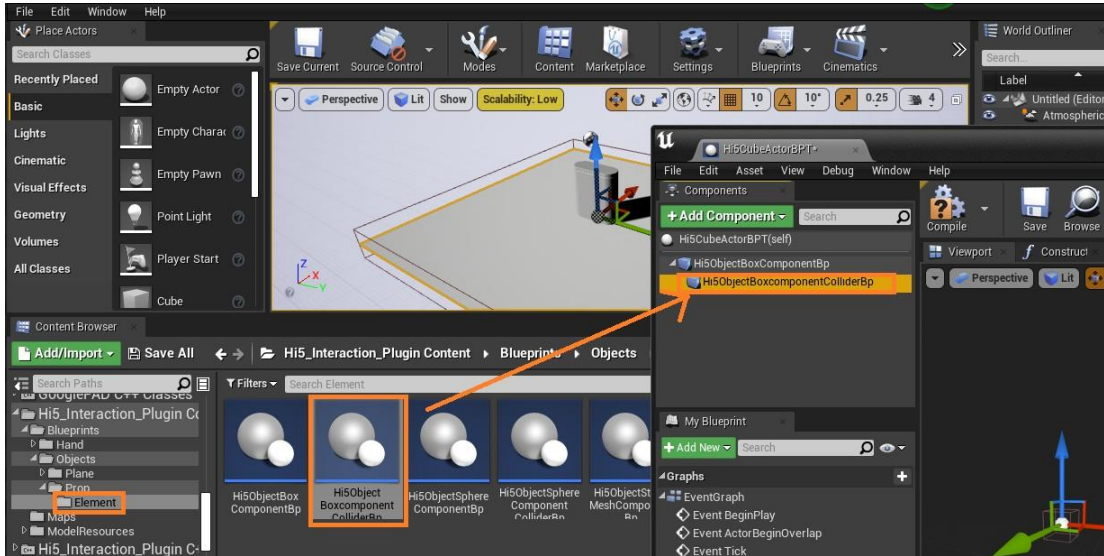
a. In the Physics panel, check Simulate Physics, as shown in the figure:



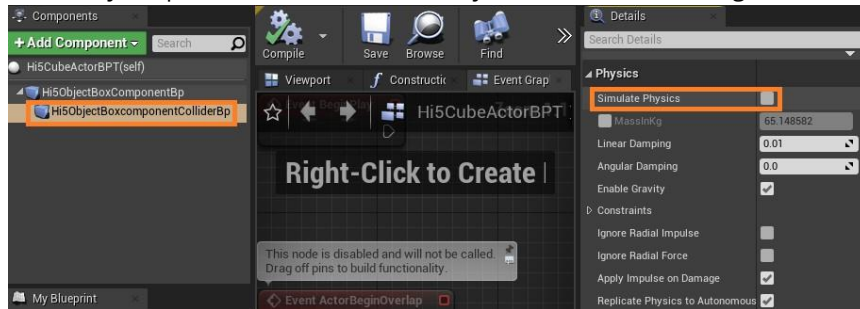
b. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object, as shown in the image:



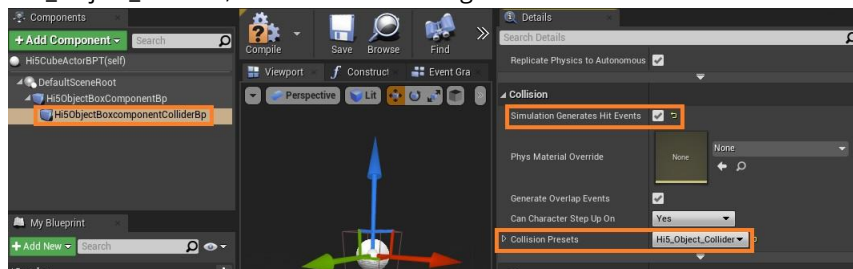
4. Drag and drop the Hi5ObjectBoxcomponentColliderBp preset in the Interactive SDK as a subclass Hi5ObjectBoxComponentBp, as shown in the figure:



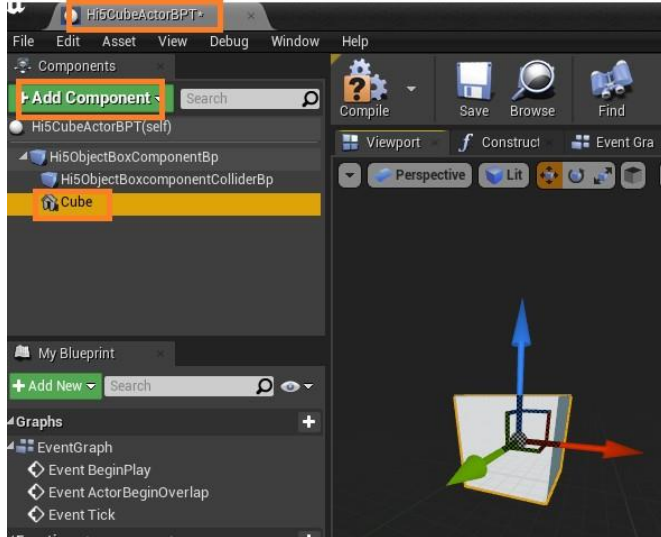
5. Hi5CubeActorBPT Select Hi5ObjectBoxcomponentColliderBp in the Blueprint class, and complete the following settings in the Details panel on the right:
 - a. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



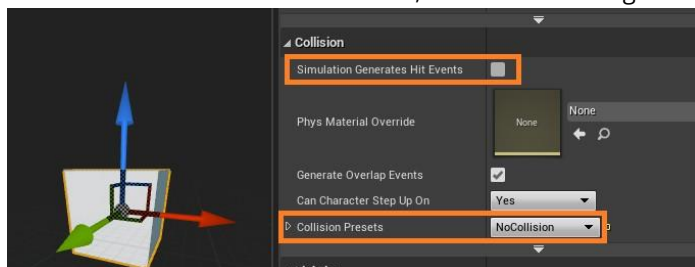
- b. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object_Collider, as shown in the image



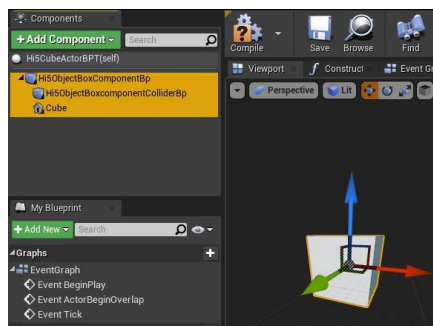
6. Select Hi5ObjectBoxcomponentBp in the Hi5CubeActBPT Blueprint, click the Add Component button, and add a Cube Static Mesh component, as shown in the figure:



7. In the Details->Collision panel on the right, Simulation Generates Hit Event is unchecked, and the Collision Presets is set to NoCollision, as shown in the figure:



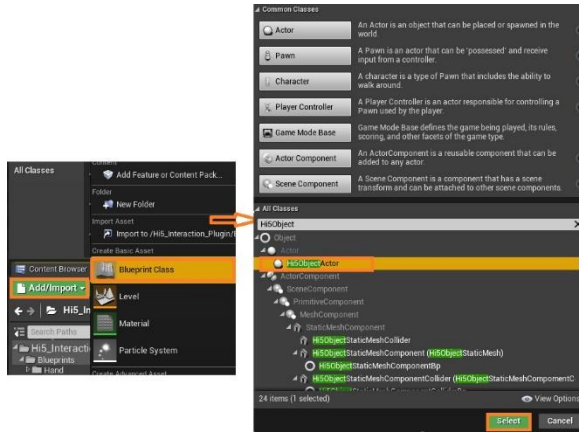
8. Adjust the Hi5ObjectBoxcompentBp, Hi5ObjectBoxcompentColliderBp and Cube sizes consistently, as shown in the figure:



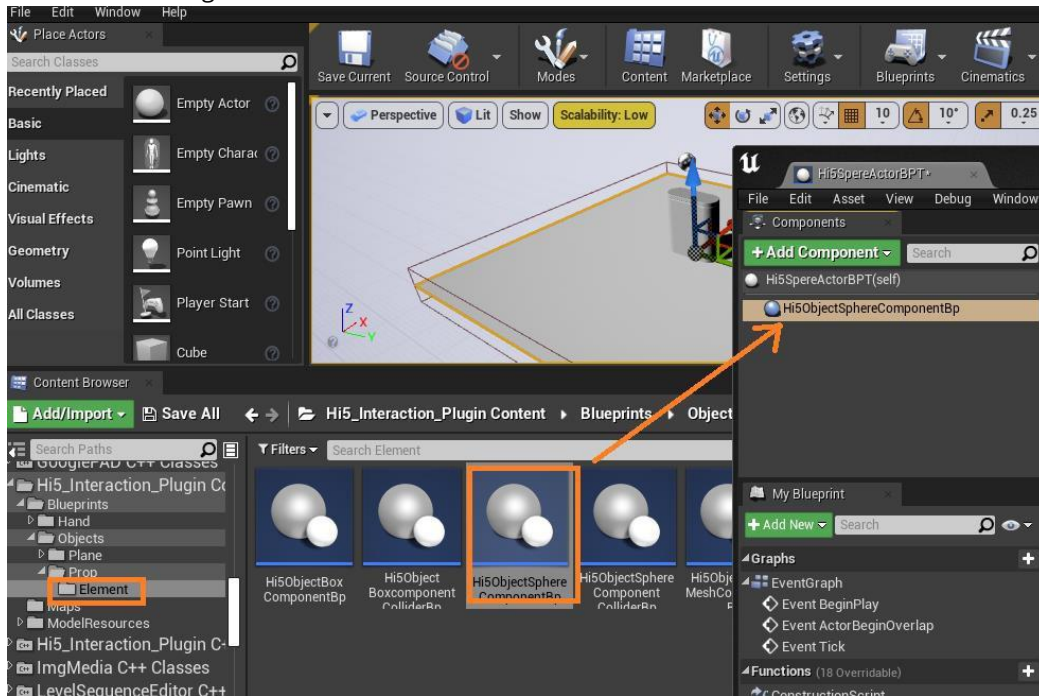
9. After the above settings are completed, click Compile and Save

Sphere

1. Create a Hi5ObjectActor Blueprint class, named Hi5SpereActorBPT here, as shown in the image:

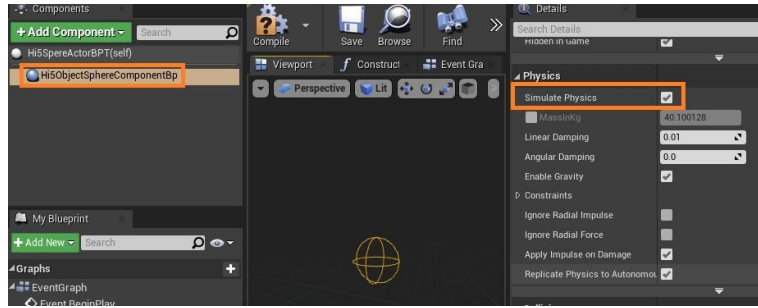


2. Double-click to open the newly created blueprint class Hi5SpereActBPT, and drag the Hi5ObjectSphereComponentBp preset in the interaction SDK to the Hi5SphereActorBp root directory, as shown in the figure:

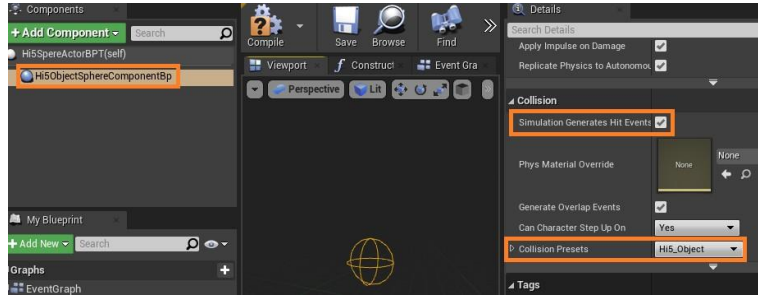


3. Hi5CubeActorBPT Select Hi5ObjectSphereComponentBp in the Blueprint class, and complete the following settings in the Details panel on the right:

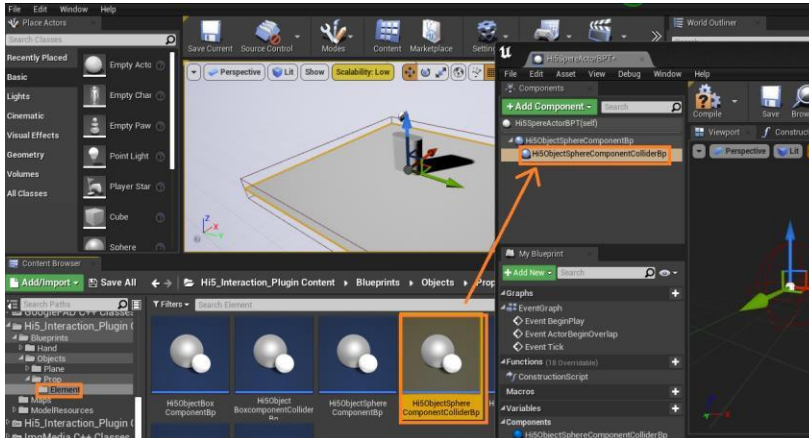
- a. In the Physics panel, check Simulate Physics, as shown in the figure:



- b. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object, as shown in the figure:

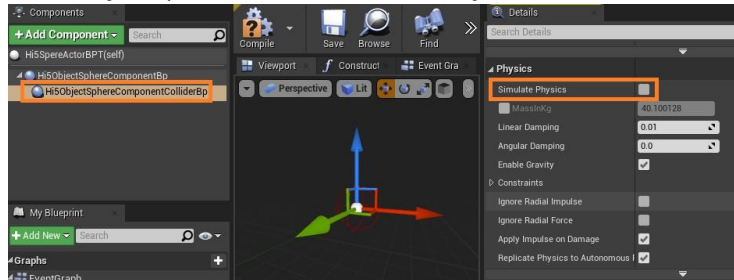


4. Drag and drop the Hi5ObjectBoxcomponentColliderBp preset in the Interactive SDK as a subclass Hi5ObjectBoxComponentBp, as shown in the figure:

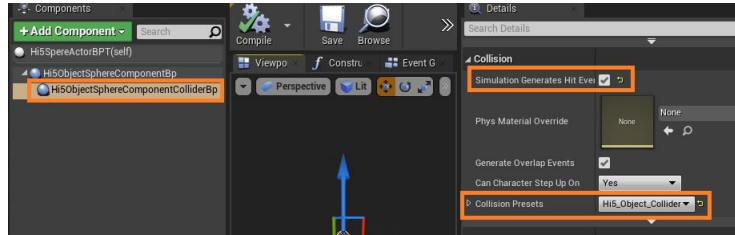


5. Hi5CubeActorBPT Select Hi5ObjectSphereComponentColliderBp in the Blueprint class, right Complete the following settings in the Details panel:

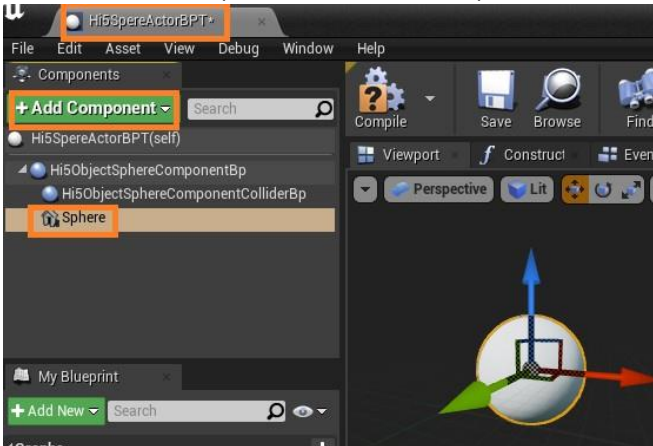
- a. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



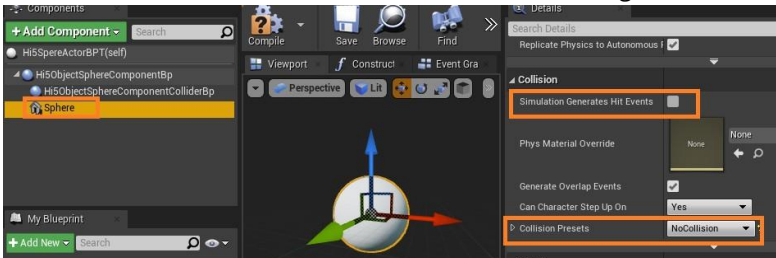
- b. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object_Collider, as shown in the figure



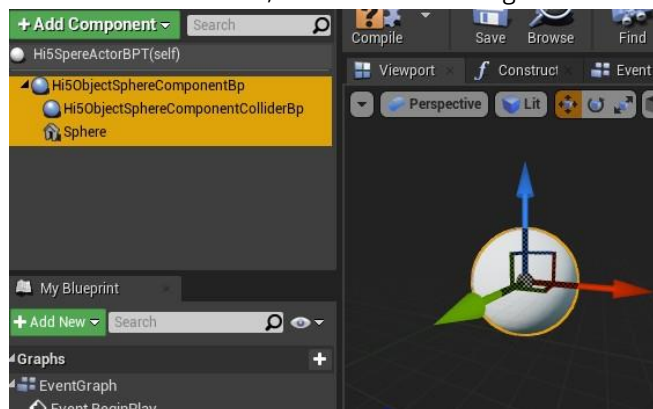
6. Select Hi5ObjectSphereComponentBp in the Hi5CubeActBPT Blueprint, click the Add Component button, and add a Sphere Static Mesh component, as shown in the figure:



7. In the Details->Collision panel on the right, Simulation Generates Hit Event is unchecked, and the Collision Presets is set to NoCollision, as shown in the figure:



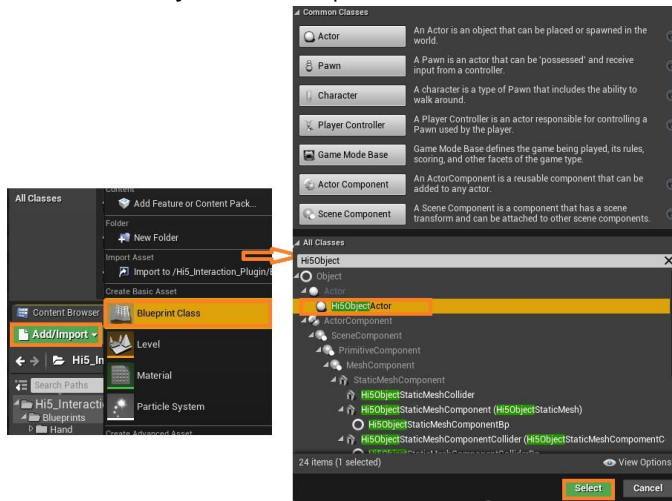
8. Adjust Hi5ObjectSpherecompentBp, Hi5ObjectSpherecompentColliderBp, and Sphere. The sizes are consistent, as shown in the image:



9. After the above settings are completed, click Compile and Save

Static Mesh

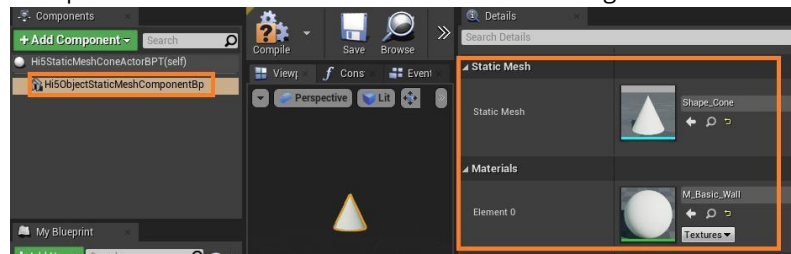
1. Create a Hi5ObjectActor Blueprint class, named Hi5StaticMeshConeActBPT, as shown in the image:



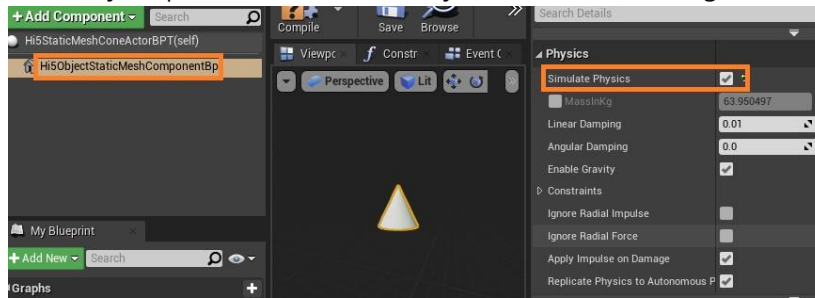
2. Double-click to open the newly created blueprint class Hi5StaticMeshConeActBPT, and drag the preset Hi5ObjectStaticMeshComponentBp in the interaction SDK to the Hi5StaticMeshConeActor BPT root directory, as shown in the figure:



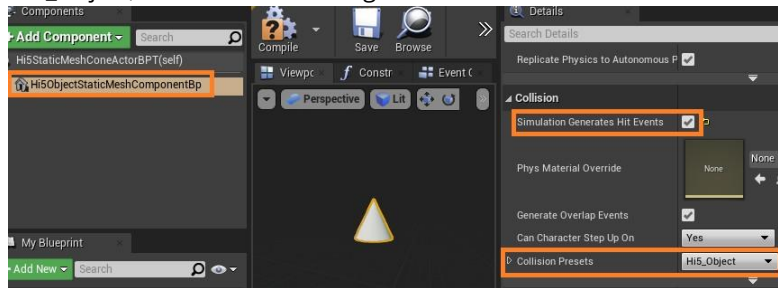
3. Hi5ObjectStaticMeshMeshComponentBp is selected in the Hi5StaticMeshMeshComponentBp in the Blueprint class, and the following settings are set in the Details panel on the right:
 - a. Set up Static Mesh and Materials as shown in the image:



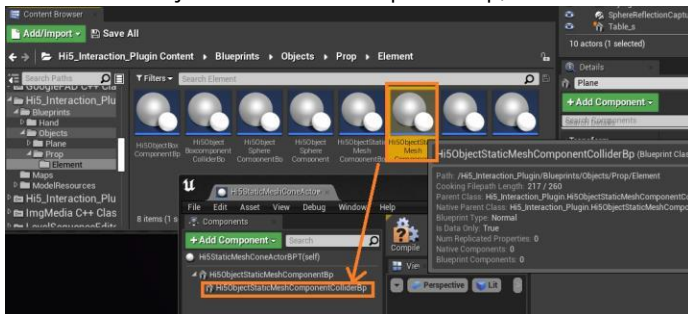
- b. In the Physics panel, check Simulate Physics, as shown in the figure:



- c. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object, as shown in the image:

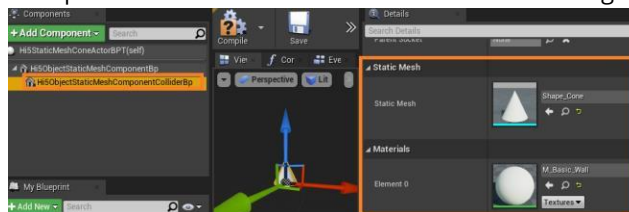


4. Drag and drop the Hi5ObjectStaticMeshComponentColliderBp preset in the Interaction SDK as a subclass Hi5ObjectStaticMeshComponentBp, as shown in the figure:

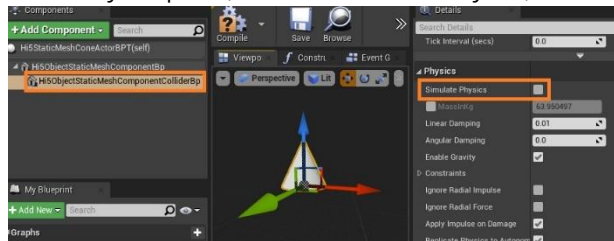


5. Hi5StaticMeshConeActorBPT Select Hi5ObjectStaticMeshComponentColliderBp in the Blueprint class, and complete the following settings in the Details panel on the right:

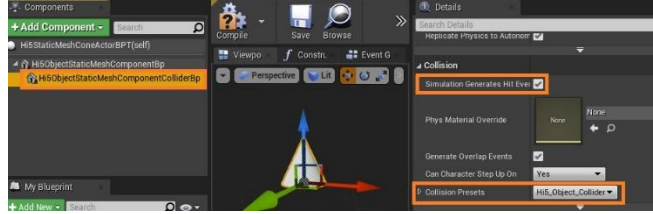
- a. Set up Static Mesh and Materials as shown in the image:



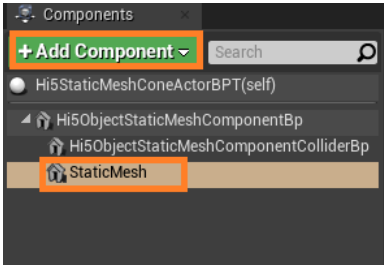
- b. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



- c. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object, as shown in the image:



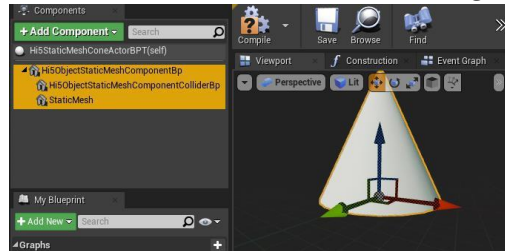
6. Hi5StaticMeshConeActor BPT Blueprint Hi5ObjectStaticMeshComponent_BP selected, click Add Component button, add a Static Mesh component, as shown in the image:



7. In the Details->Collision panel on the right, Simulation Generates Hit Event is unchecked, and the Collision Presets is set to NoCollision, as shown in the figure:



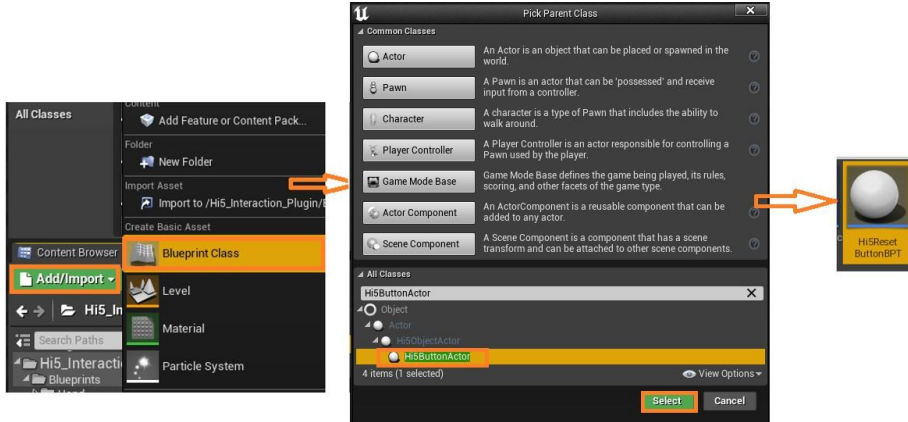
8. Adjust Hi5ObjectStaticMeshComponentBP, Hi5ObjectStaticMeshComponentColliderBP The size of the static mesh is the same, as shown in the figure:



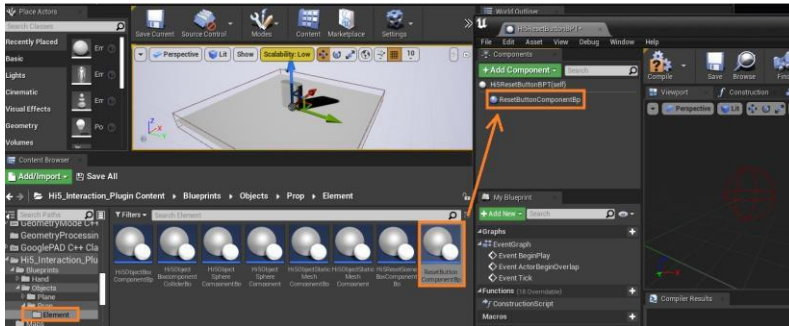
9. After the above settings are completed, click Compile and Save

Reset Button

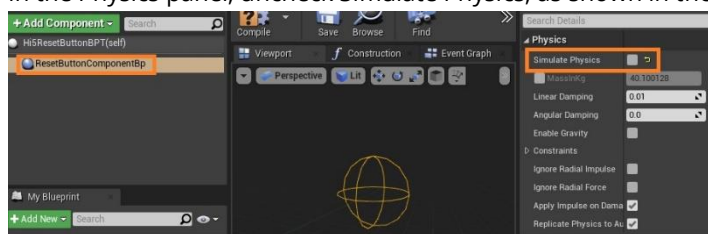
1. Create a Hi5ButtonActor Blueprint class, named Hi5ResetButtonBPT, as shown in the image:



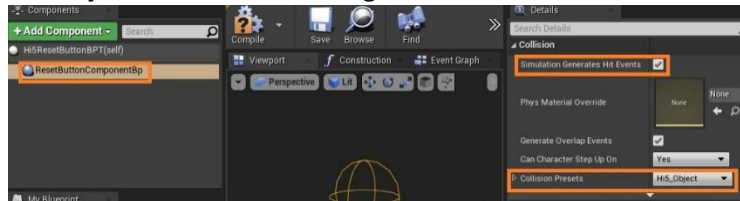
2. Double-click to open the Blueprint class you just created, Hi5ResetButtonBPT, which will interact with the prebuilt SDK ResetButtonCompentBp as a subclass dragged to the root directory of Hi5ResetButtonBPT, as shown in the figure:



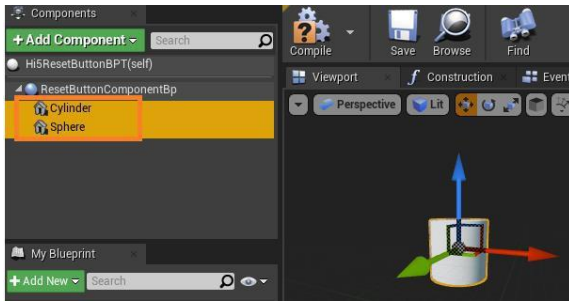
3. Select ResetButtonCompentBp and complete the following settings in the Details panel on the right:
 - a. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



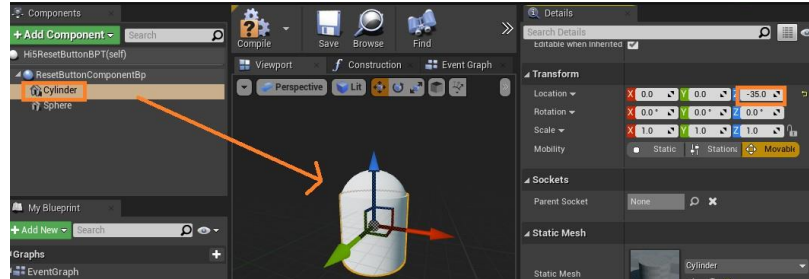
- b. Check Simulation Generates Hit Event in the Collision panel, and set Collision Presets to Hi5_Object, as shown in the image:



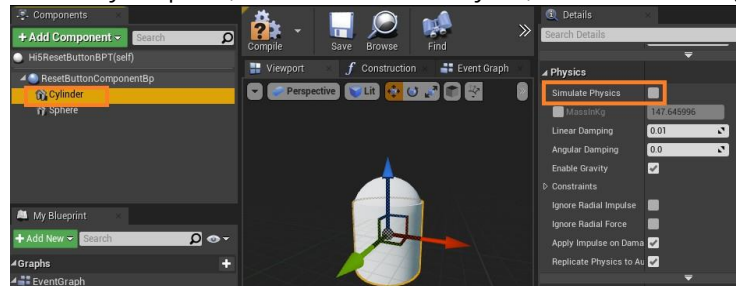
4. Select ResetButtonCompentBp, click the Add Component button, add Cylinder, and then add Sphere Two static meshes, as shown in the image:



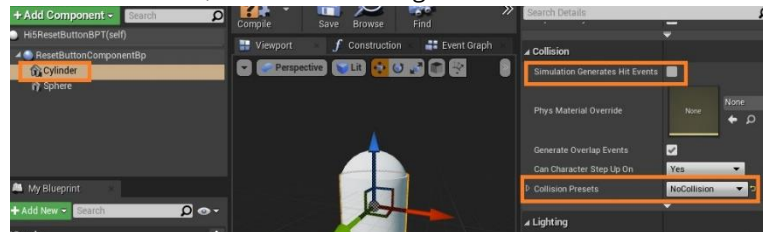
5. Select Cylinder and complete the following settings in the Details panel on the right:
 - a. In the Transform panel, adjust the position relationship as shown in the figure:



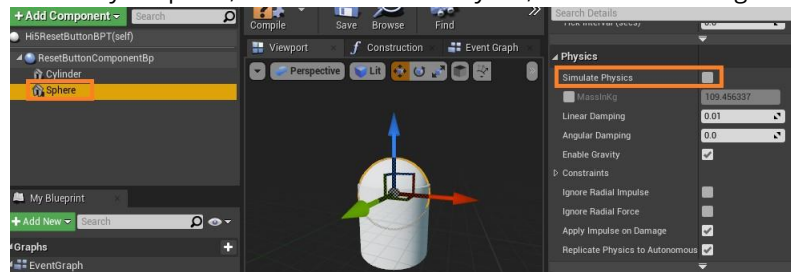
- b. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



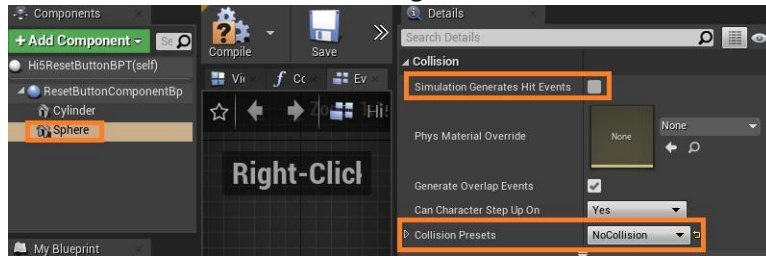
- c. The Collision panel does not check Simulation Generates Hit Event, and the Collision Presets is set to NoCollision, as shown in the figure:



6. Select Sphere and complete the following settings in the Details panel on the right:
 - a. In the Physics panel, uncheck Simulate Physics, as shown in the figure:



- b. The Collision panel does not check Simulation Generates Hit Event, and the Collision Presets is set to NoCollision, as shown in the figure:

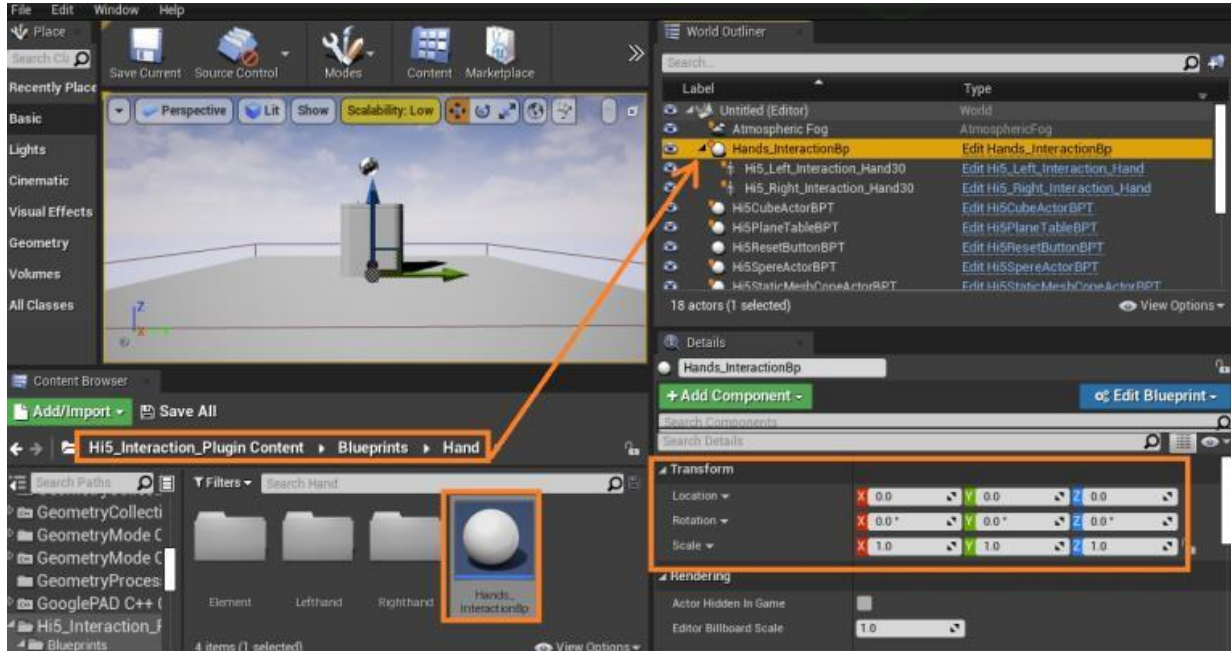


7. After the above settings are completed, click Compile and Save

Interaction Scene Setting

Hand model settings

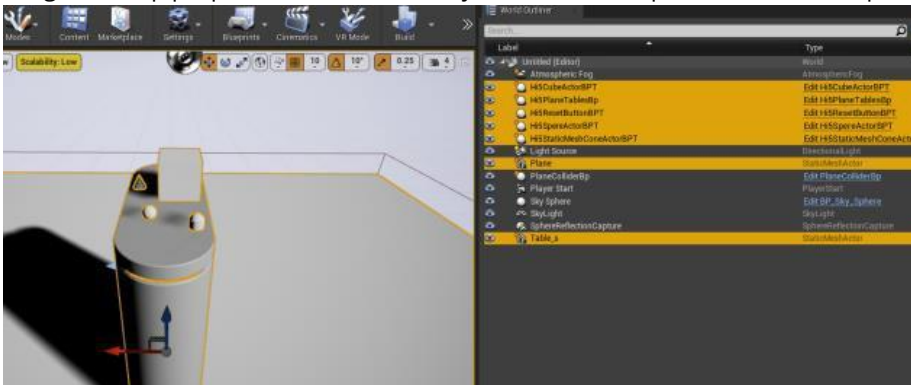
Drag and drop the preset Hands_InteractionBp Blueprint from the Interactive SDK into the scene, and the Transform Location Rotation is reset to 0 and the scale is set to 1, as shown in the figure:



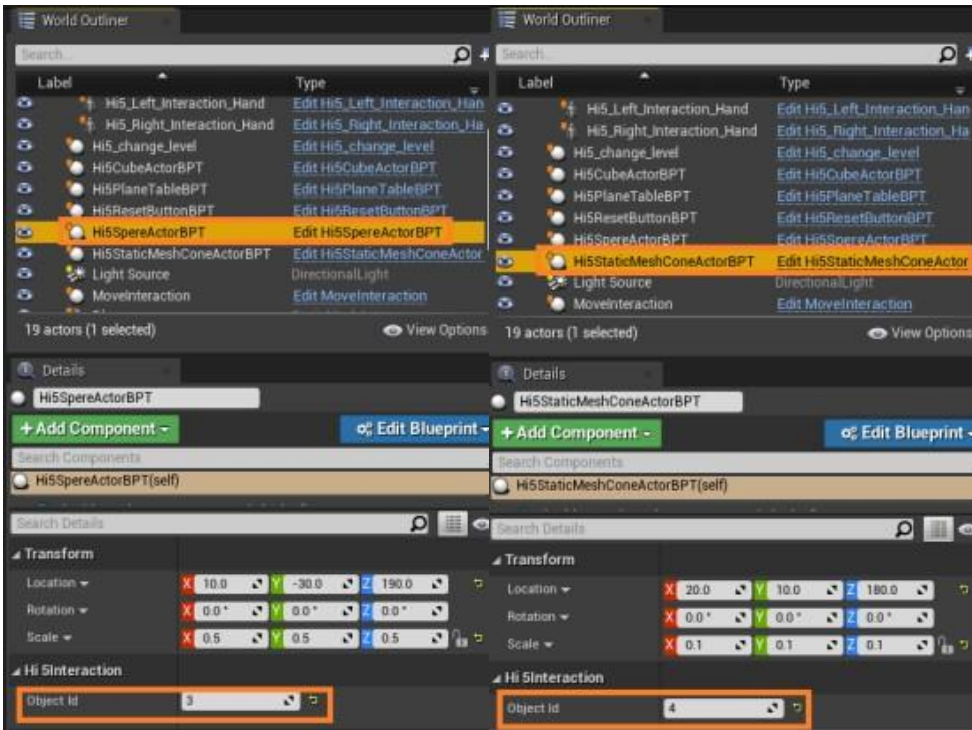
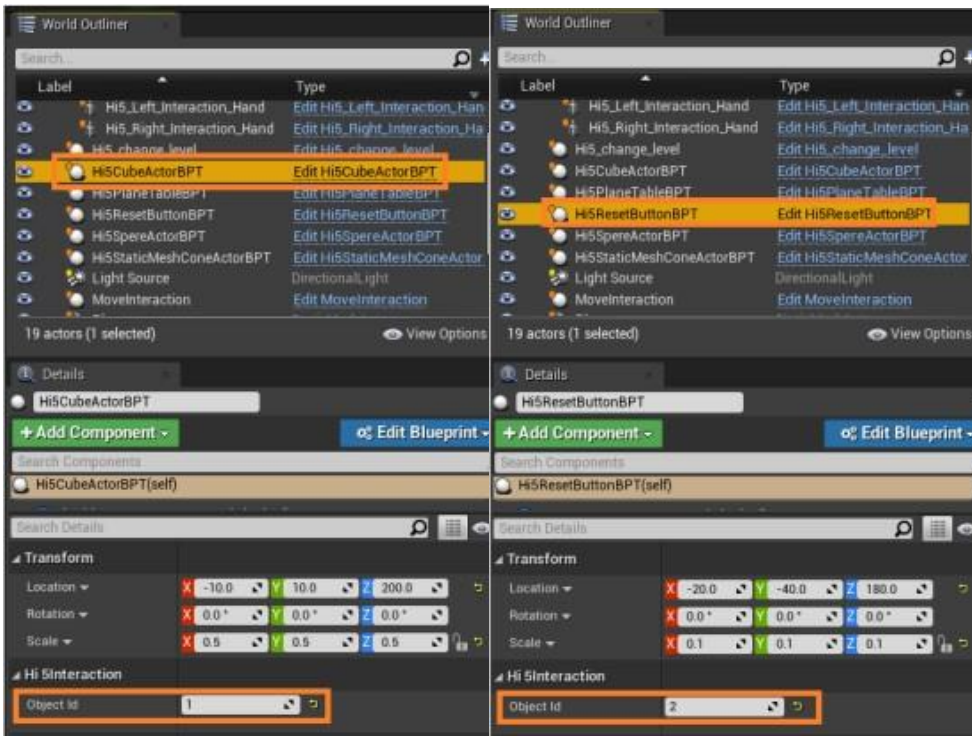
*Note: If the user needs to modify the hand model, the specific operation is more complicated, please contact after-sales.

Prop and headset settings

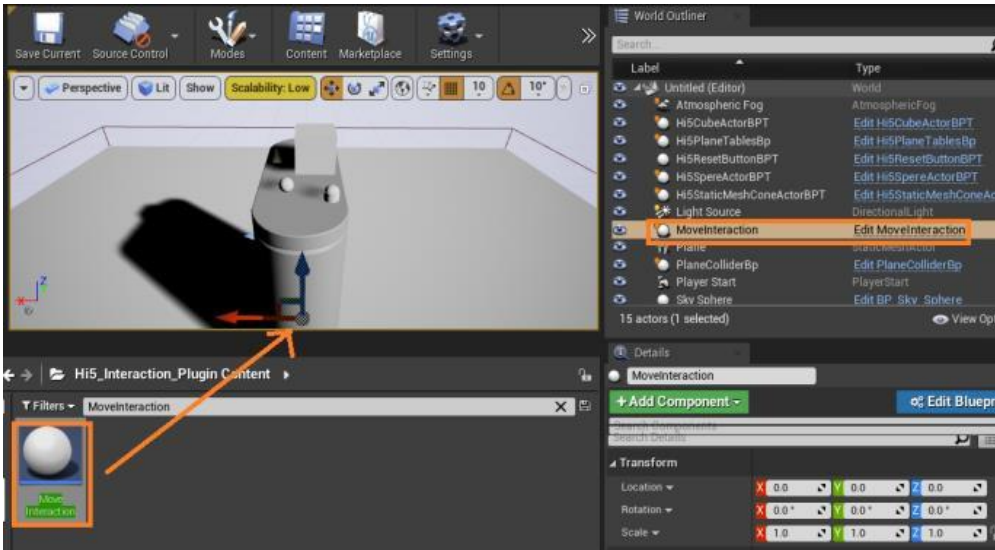
1. Drag and drop props into the scene to adjust the size and position relationship, as shown in the figure:



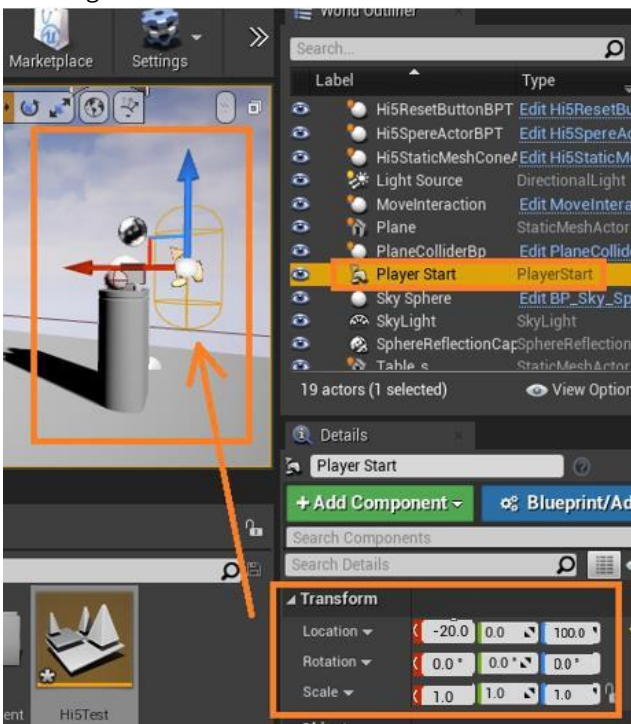
2. Select the props in World Outliner and set the Object ID in the Details->Hi5Interaction panel, as shown in the figure:



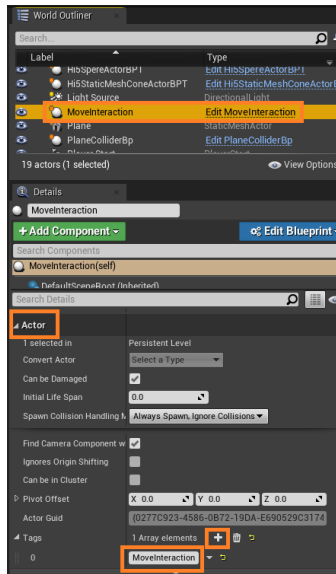
3. Drag the MoveInteraction Blueprint into the scene as shown in the image:



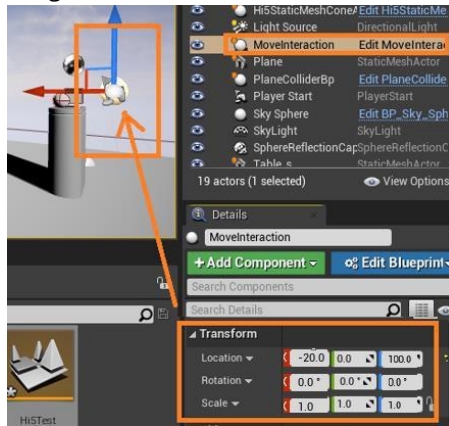
4. Check Player Start in World Outliner and set the headset start position in Details->Transform, as shown in the figure:



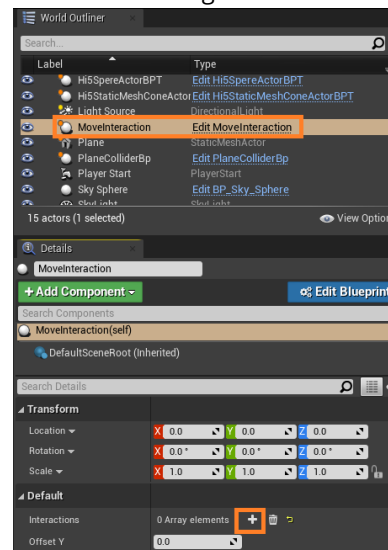
5. Select MoveInteraction in World Outliner and complete the following settings in the Details panel:
 - a. Click the + button in the Actor panel to add a Tag, then enter MoveInteraction, as shown in the image:



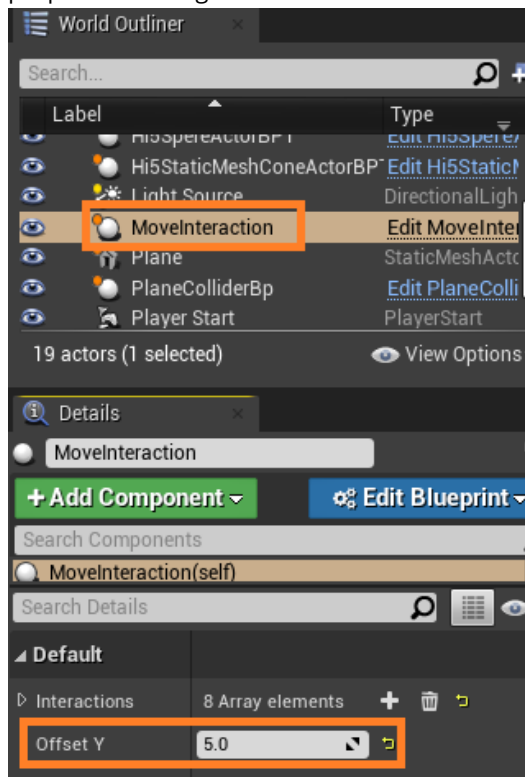
- b. Assign the Transform value that Player Start just set to MoveInteraction, as shown in the image:



- c. Click the + button in the Default panel to add the interactive props you just added to the scene as shown in the figure:



- d. Set Offset Y, which sets the position relationship between the height of the headset and the prop after clicking the Reset button on the back of the hand, as shown in the figure:

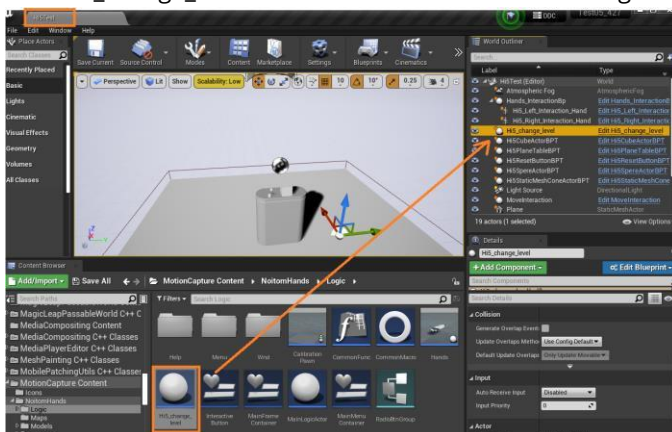


*Note: Click the back of the left hand with the right index finger during the run, and the Reset key resets the height of the component in the scene

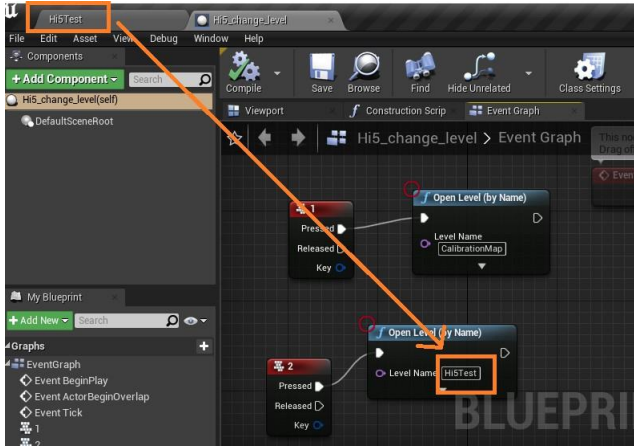
*Note: In this example, the headset is calibrated to 80cm

Scene switching settings

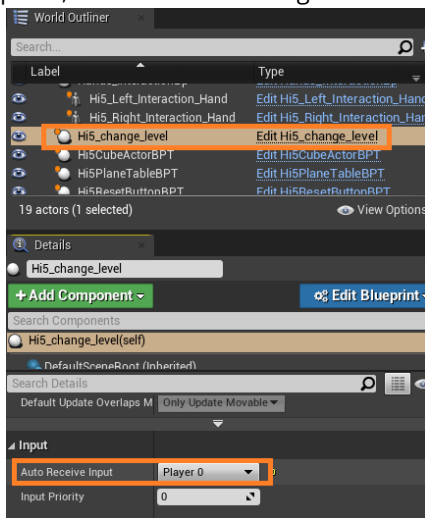
1. Add Hi5_change_level to the scene as shown in the image:



2. Open the Hi5_change_level Blueprint, enter the name of the newly created interaction scene, shortcut key 1 to switch to calibration scene, that is, CalibratnionMap, shortcut key 2 to switch to interaction scene, that is, Hi5Test as shown in the figure:



3. Check Hi5_change_level in World Outliner and set Auto Receive Input to Player0 in the Detail->Input panel, as shown in the image:



Appendix: Interactive SDK API Interface Description

C++

The Hi5InteractionInterfaceMessage.h file provides the relevant SDK interface. The specific path is under the plugin Source\Hi5_Interaction_Plugin\Public\utilities\Hi5InteractionInterfaceMessage.

```
class HI5_INTERACTION_PLUGIN_API Hi5InteractionInterface
{
public:
virtual void ChangeHandState(FHi5HandStateDataInterface handData) = 0; virtual void
ChangeHandGestureState(FHi5HandGestureStateDataInterface handGestureData) =0;
virtual void ChangeObjectState(FHi5ObjectStateDataInterface stateData) = 0; virtual
void ChangeButtonState(FHi5ButtonStateDataInterface stateData) = 0;
};

class HI5_INTERACTION_PLUGIN_API Hi5InteractionInterfaceMessage
{
public:
static Hi5InteractionInterfaceMessage& GetInstance() { static
Hi5InteractionInterfaceMessage instance; return instance;
}
void RegisterInterface(Hi5InteractionInterface* item); void
UnRegisterInterface(Hi5InteractionInterface* item);
};
```

How to use:

The user creates a new class that inherits the Hi5InteractionInterface class and implements 4 pure virtual functions of the class that are called when the class starts executing

```
Hi5InteractionInterfaceMessage::GetInstance().RegisterInterface(
Hi5InteractionInterface* item)*register. Called when the class finishes executing
Hi5InteractionInterfaceMessage::GetInstance().UnRegisterInterface(Hi5InteractionI
nterface* item)unregister
void ChangeHandState(FHi5HandStateDataInterface handData)
{
FHi5HandStateDataInterface handData;
{
EHi5HandState HandState; enum EHi5HandState

{
ERelease = 0, //no operational status EPinch, //grab object state
EPinch2, //Two fingers pinch the object state
EFlyPinch, //The state of grabbing in the air has not been turned on for the time
being
ELift, //The state of lifting objects EPat, //The state of hitting the object

};
int ObjectID; //In the state of grasping and lifting, this value is meaningful and
represents the object to be manipulated Id. bool isLeft; //Is it left handed
```

```

int peopleId;
}
}
void ChangeHandGestureState(FHi5HandGestureStateDataInterface handGesturData)
{
FHi5HandGestureStateDataInterface handGestureData;
{
EHi5HandGestureState HandState;
{s
Non = 0,
Fist = 1, //First State FingerPlane = 2, //Stretched fingers HandIndexPoint = 3,
//index finger out
OK = 4, // OK state
}
}
}
void ChangeObjectState(FHi5ObjectStateDataInterface stateData)
{
FHi5ObjectStateDataInterface stateData
{
int ObjectID;// object Id EObjectState state;//
{
ENon = 0, //idle state EPinch = 1, //crawl status EStatic, //Stationary
State E_lift, //lifting status EMove, //Mobile state
EPat //slap state
}
bool isButton; //Whether it is a button
int peopleId; //Grab, lift, slap state operator Id

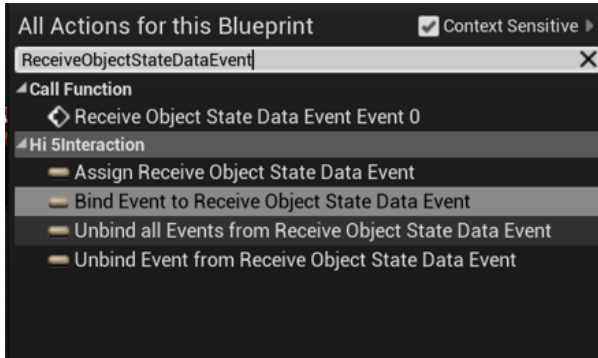
bool isLeft; //Grab, lift, slap the state of the operator left and right hand to judge
}
}
void ChangeButtonState(FHi5ButtonStateDataInterface stateData)
{
FHi5ButtonStateDataInterface stateData
{
int ObjectID;// object Id EButtonState state;
{
ENon = 0, //
EStatic, //Stationary state
EPat //Slap status
}
int peopleId; //Grab, lift, slap state operator Id
bool isLeft; // Grab, lift, and slap the operator's left and right hands to judge

}
}
}

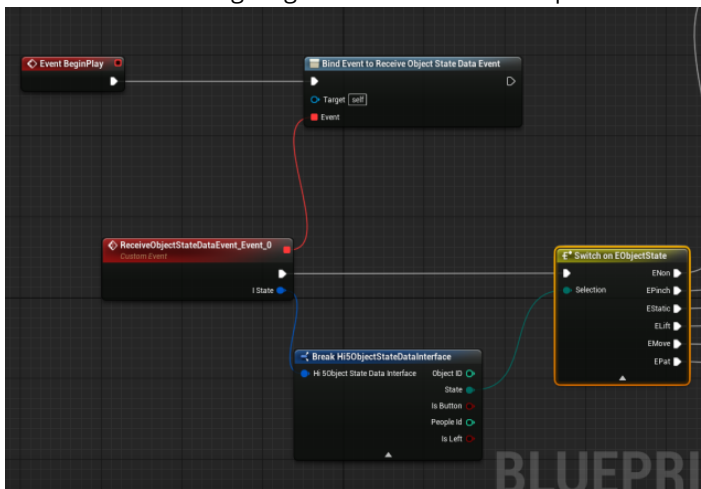
```

Blueprint interface

1. Hi5ObjectActor Interactive Object Blueprint binding interface, call the interface when the state of the object changes, right-click in Hi5ObjectActor EventGraph to enter ReceiveObjectStateDataEvent, and select Bind Event to Receive Object State Data Event, as shown in the figure:



Refer to the following diagram to outline the blueprint:



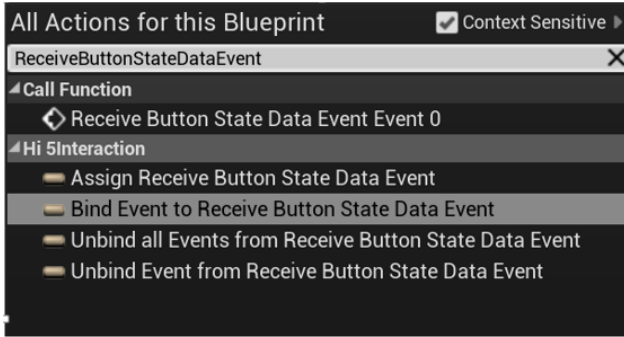
```

EObjectState state;
{
ENon = 0, //Idle
EPinch = 1, //Grab State
EStatic, //Stationary
ELift, //Hold State
EMove, //Move State
EPat // Slap State
}

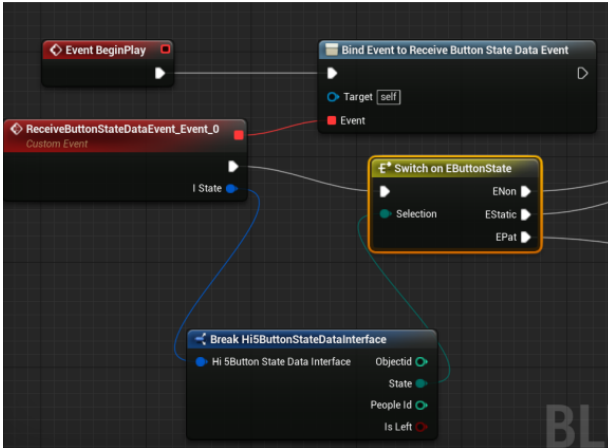
```

2. Hi5ButtonActor interactive button blueprint binding interface, call the interface when the state of the button changes, right-click in the Hi5ButtonActor EventGraph to enter ReceiveButtonStateDataEvent,

select Bind Event to ReceiveButtonStateDataEvent, as shown in the figure:



Refer to the following diagram to outline the blueprint:



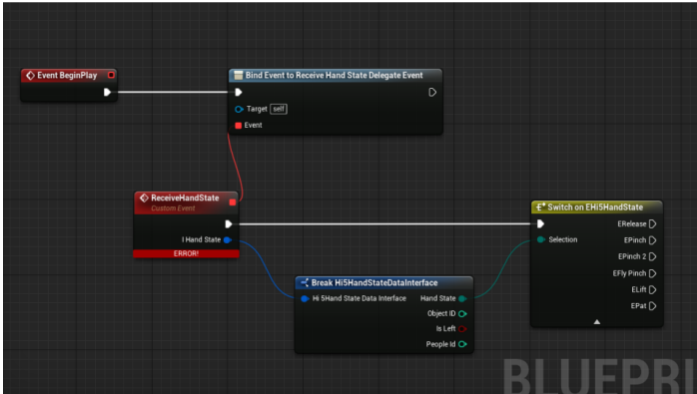
EButtonState state;

```
{
ENon = 0, //
EStatic, //stationary state
EPat // Slap status
}
```

- Hi5SkeletalMeshActor interactive hand state update and pose state update in Hi5SkeletalMeshActor Input ReceiveHandStateDelegateEvent under EventGraph, this interface returns hand state



Draw a blueprint with reference to the figure below:



EHi5HandState

```

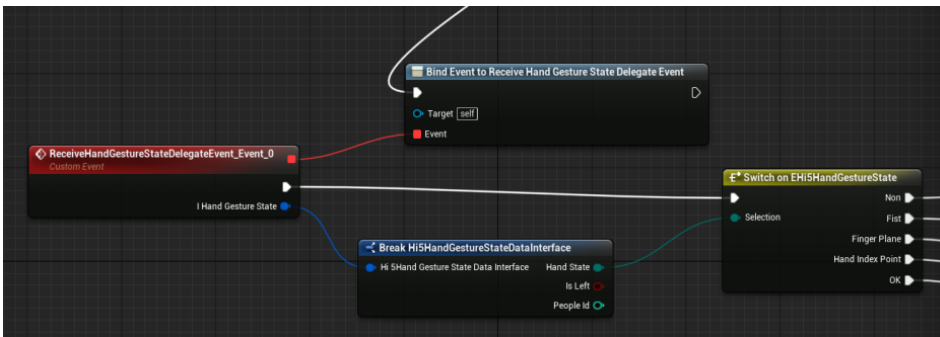
{
ERelease = 0, //No operating state
EPinch, //Gripping object state
EPinch2, //Pinch object state with two fingers
EFlyPinch, //Air Grab Status is not turned on at the current state.
ELift, //Lifting Object State
EPat, //Slap Object State
};

```

Enter ReceiveHandGestureStateDelegateEvent under Hi5SkeletalMeshActor EventGraph

The interface returns the hand gesture recognition status.

Refer to the following diagram to outline the blueprint:



EHi5HandGestureState

```

{
Non = 0,
Fist = 1, //fist state
FingerPlane = 2, //Finger flat
HandIndexPoint = 3, //index finger extended state
OK = 4, // OK state
}

```