HI5 2.0 INTERACTION SDK USER MANUAL

STEAM VR PLATFORM HEADSET SERIES HTC VIVE PRO / PIMAX

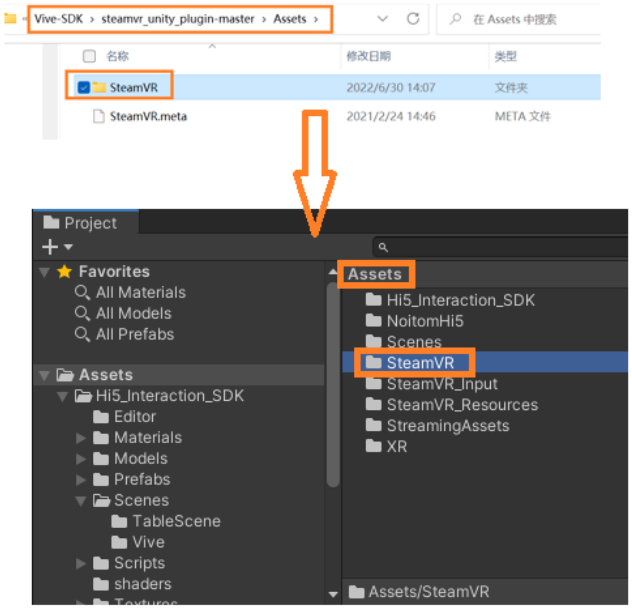
Contents

Unity VR basic environment configuration 2
Interaction SDK Installation 4
Interaction SDK app 5
1. Project Settings 5
2. Scene Settings 6
3. Object Settings 8
4. Button Settings 10
Related interfaces 11
1. Hand-related interfaces 11
I. Hand state 11
II. Hand gesture recognition status 11
2. Hand Event Interface 11
3.Interactive Object interface 12
4. Button Interface 13

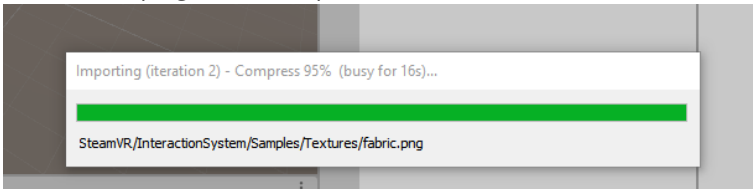
Unity VR basic environment

Please use Unity 2019.4.18 or above to create a new project.

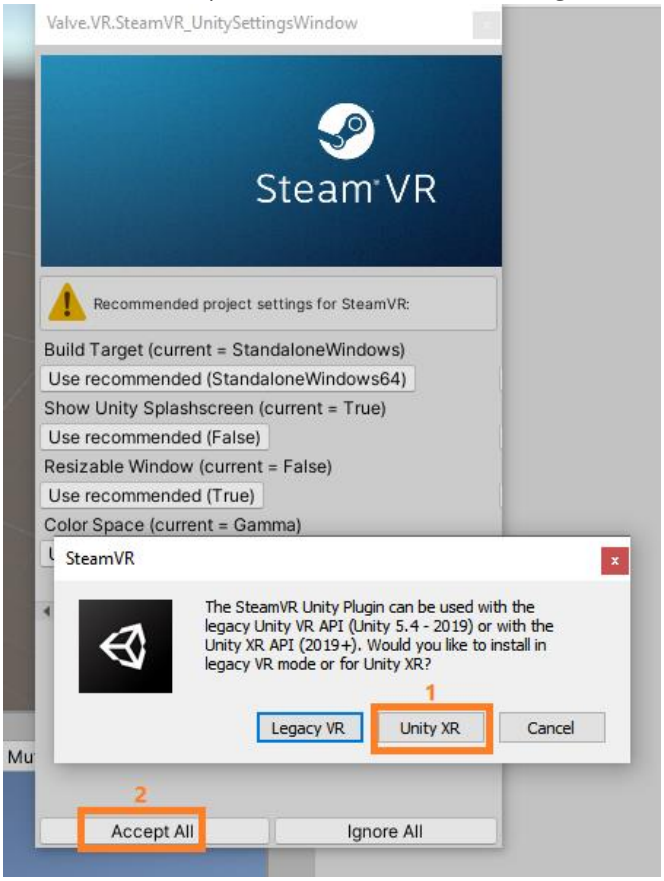
1. Plugin Download
   Go to the official Steam website https://github.com/ValveSoftware/steamvr_unity_plugin to download the latest version of Steam VR Unity plugin.
2. Plugin installation
   a. Create or open a new Unity project.
   b. Open the plugins folder and copy the SteamVR directory from the Assets folder to the Assets folder in your Unity project, as shown in the figure:



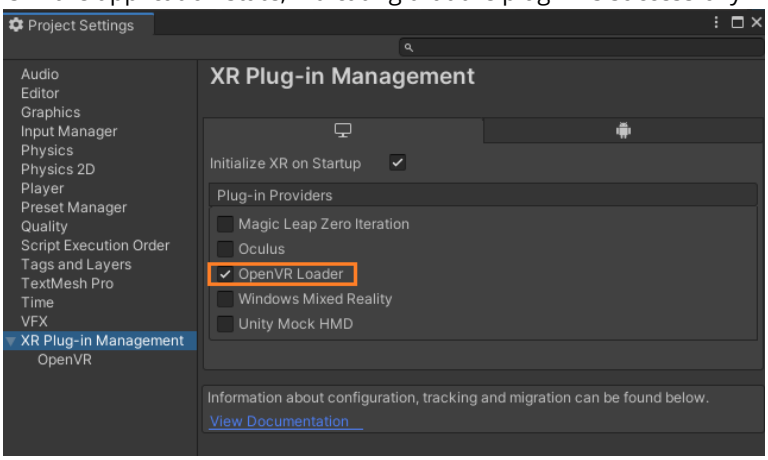
3. Wait for the plug-in to be imported



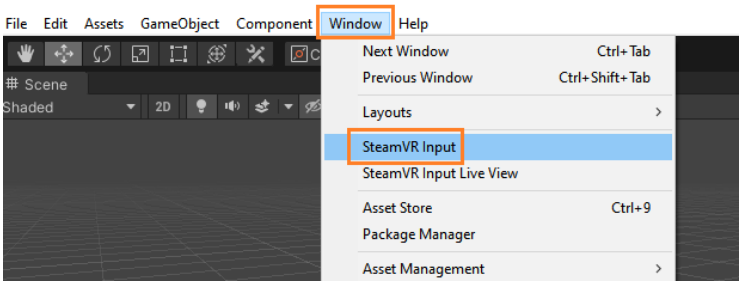
4. After the SteamVR plug-in is successfully imported, click the UnityXR button in the pop-up window, and then click the Accept All button, as shown in the figure:



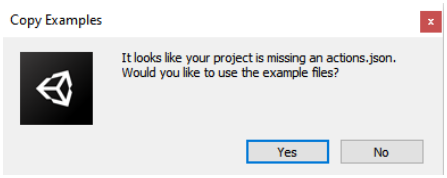
5. After the installation is complete, Project Settings will pop up, showing that the OpenVR Loader plug-in is in the application state, indicating that the plug-in is successfully installed, as shown in the figure:

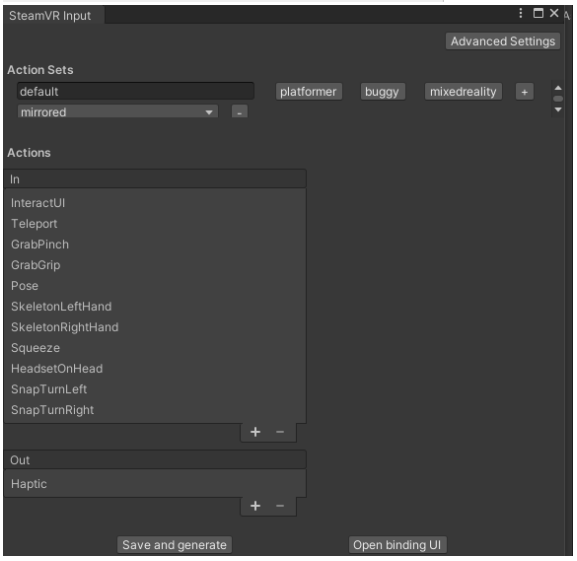


After completing the above configuration, you can find the SteamVR Input button in the window, click Windows ->SteamVR Input



If the Copy Examples pop-up window appears, click the Yes button, and then wait for the import and compilation.





Interaction SDK Installation

Unity VR environment configuration is complete, first install Hi5 2.0 base SDK, and then install the interaction SDK.

1) first import Hi5 2.0 base SDK :)

   Hi5 2.0_SteamVRHeadset_FSDK_Unity_v1.1.0.23

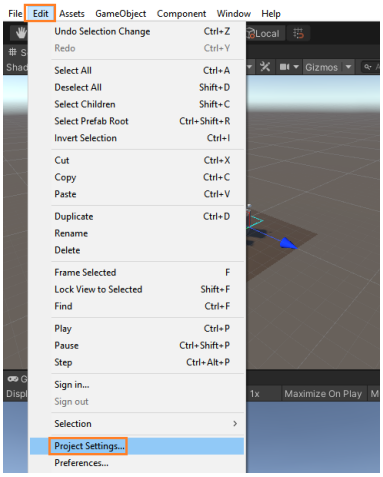
2) then import the interaction SDK:

   Hi5 2.0_SteamVRHeadset_ISDK_Unity_1.1.0.3

Interaction SDK app

1. Project Settings

Click Edit -> Project Settings to open the project settings window, as shown in the figure:



1.1 Set Tags and Layers

As shown in the figure:

Layer 8 Hi5OtherFingerTail

Layer 9 Hi5OtherFingerOther

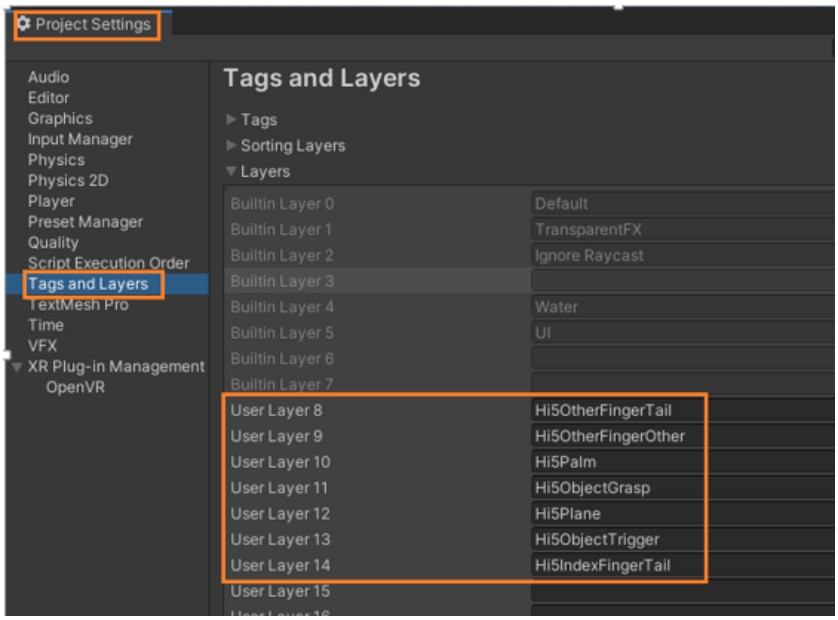
Layer 10 Hi5Palm

Layer 11 Hi5ObjectGrasp

Layer 12 Hi5Plane

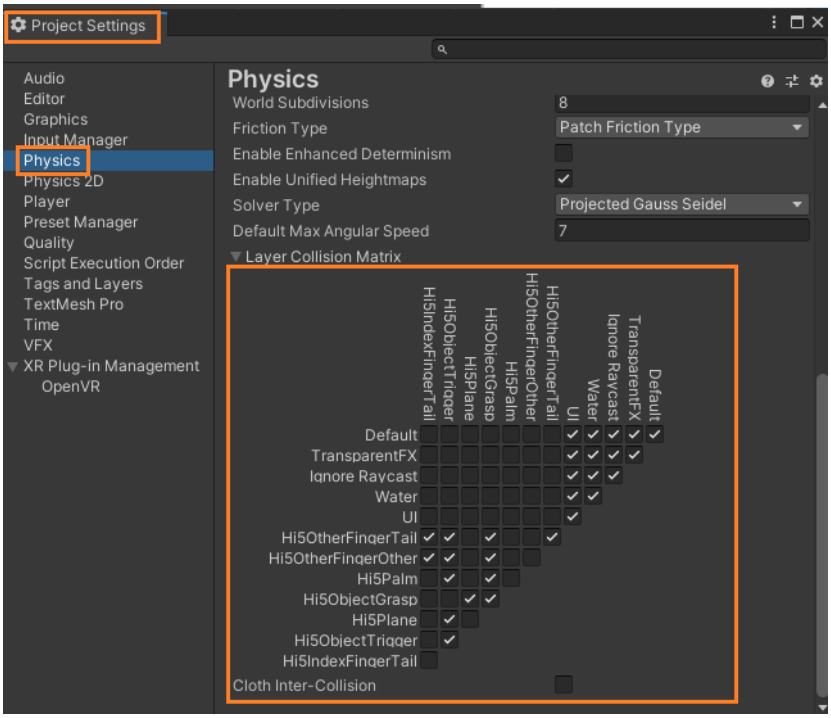
Layer 13 Hi5ObjectTrigger

Layer 14 Hi5IndexFingerTail



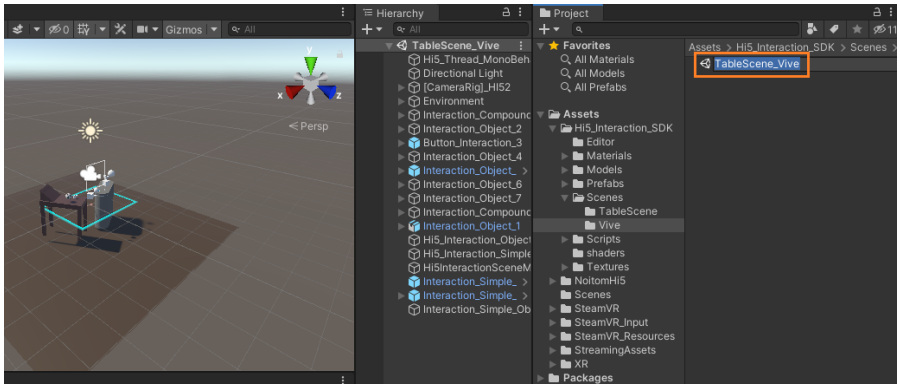
1.2 Setting up Physics.

As shown in the figure:



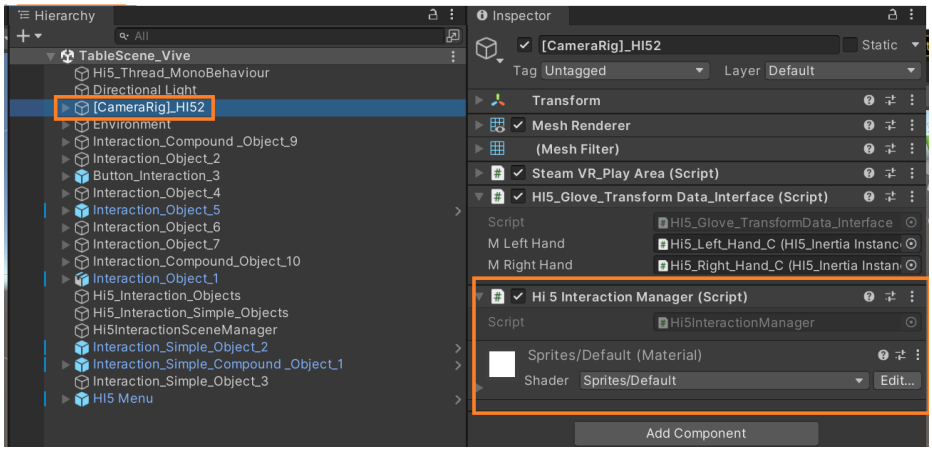
2. Scene Settings

Open the sample scene TableScene_Vive and refer to its settings as shown in the figure:

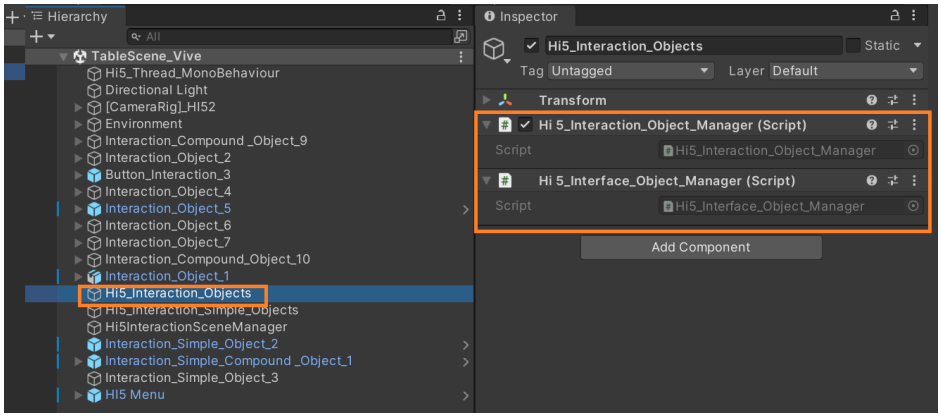


The scene must contain the following:

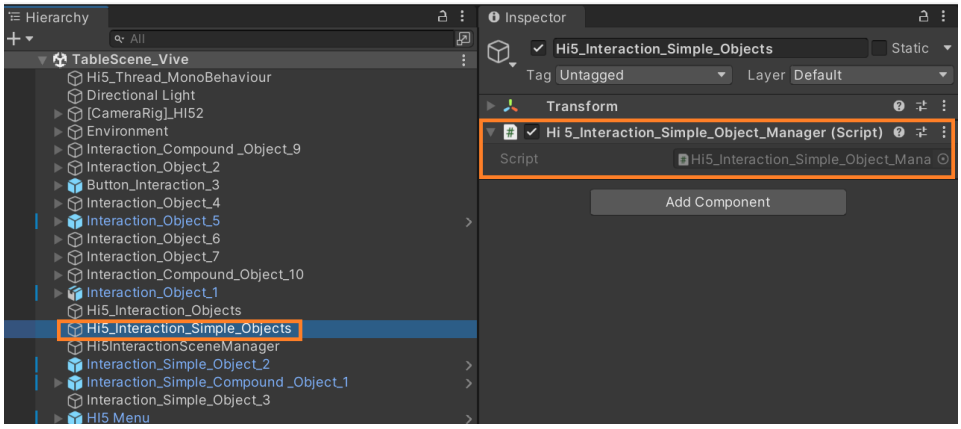
1) Hi5 Interaction Manager



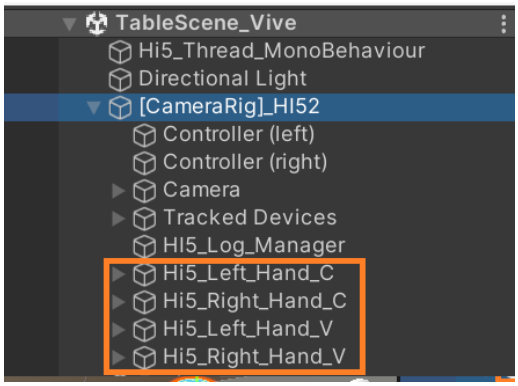
2) Hi5_Interaction_Objects



3) Hi5_Interaction_Simple_Objects



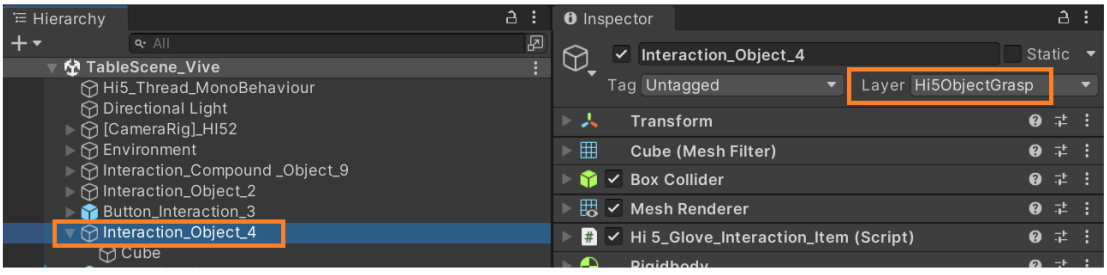
4) Hi5_Left_Hand_C , Hi5_Right_Hand_C , Hi5_Left_Hand_V , Hi5_Right_Hand_V

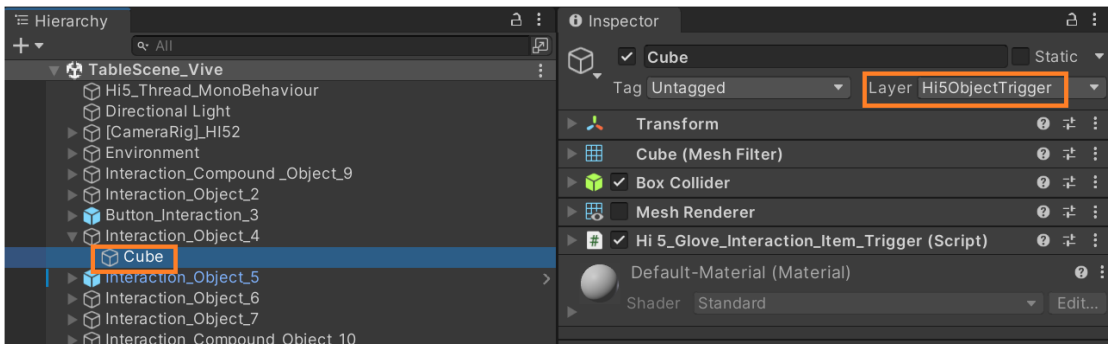


3. Object Settings

3.1 General interactive object settings

Object settings are divided into parent object and child object settings, the parent object Layer to be set to Hi5ObjectGrasp, child object Layer to be set to Hi5ObjectTrigger, such as Interaction_Object_4 -> Cube, set as shown in the figure:

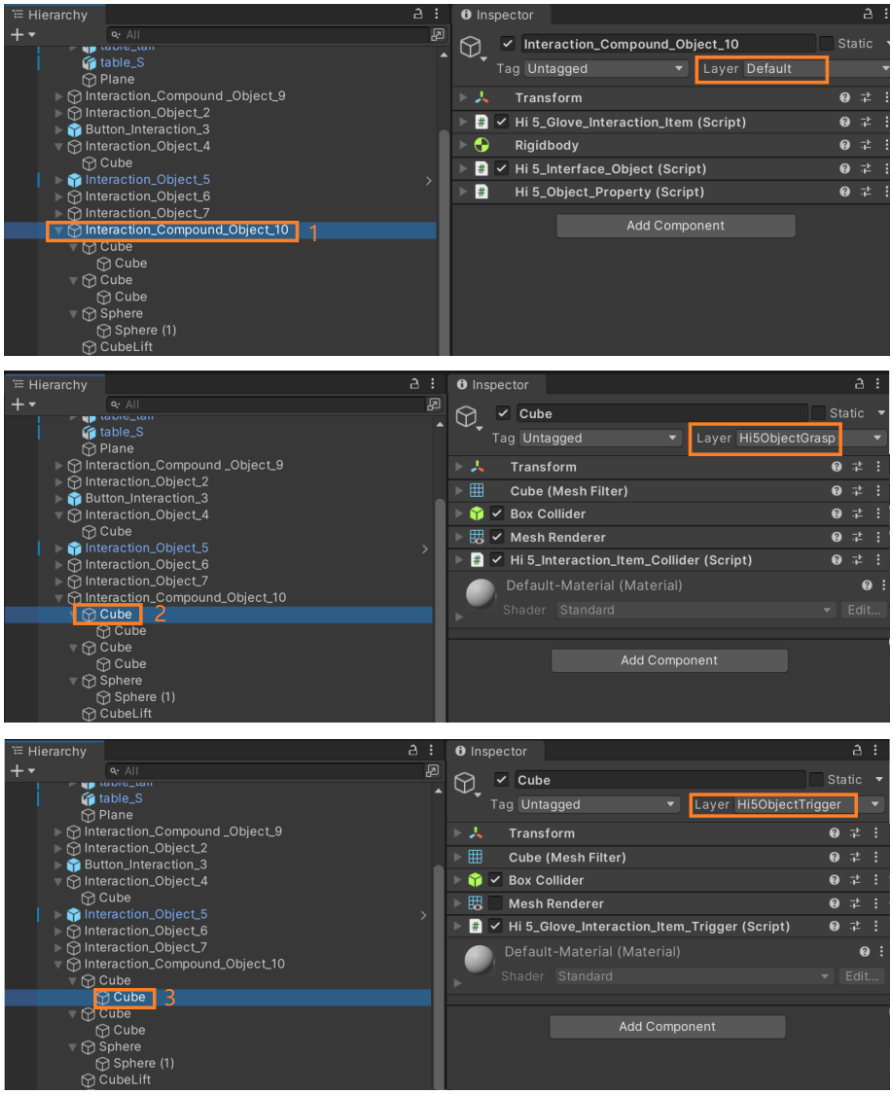




3.2 Combined object settings

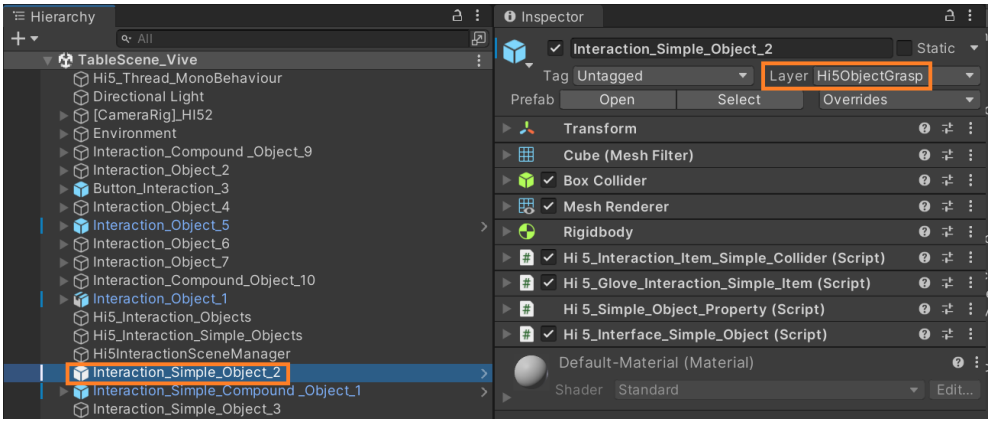
Combined objects are divided into three layers, the outermost object Layer of the combined object is set to Default, the outer layer of the combined object is set to Hi5ObjectGrasp, the inner layer of the combined object is set to Hi5ObjectTrigger.

For example, Interaction_Compound_Object_10, set as shown in the figure:



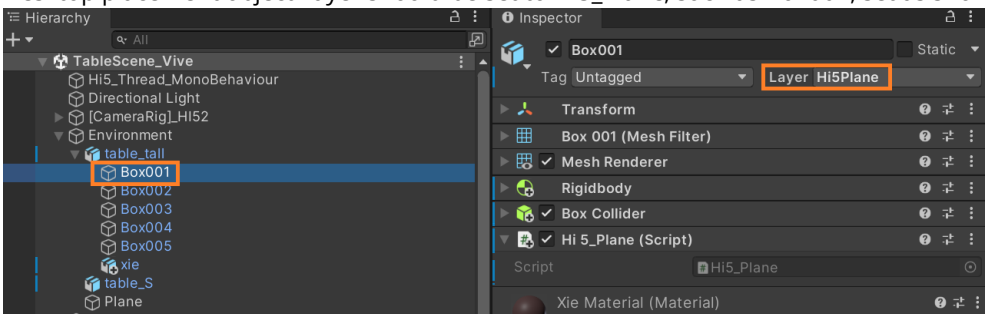
3.3 Simple object settings

Simple objects only grasping and other functions, they do not produce movement, when grabbed and released will stay in the original position, its Layer should be set to Hi5ObjectGrasp, such as Interaction_Simple_Object_2, set as shown in Figure:



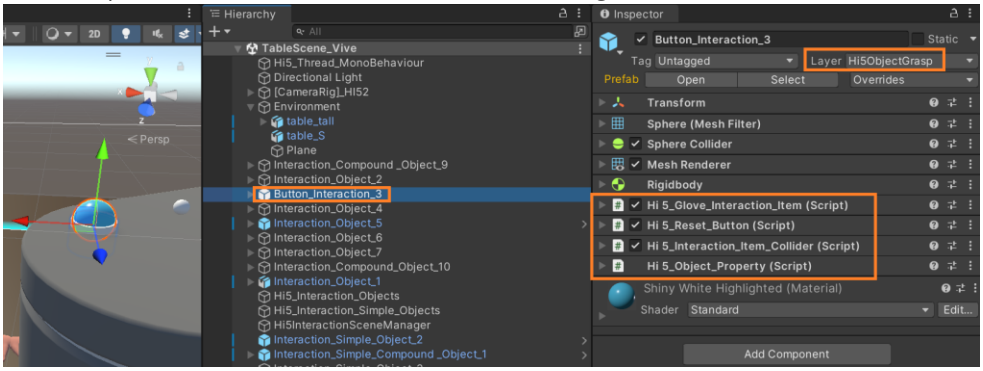
3.4 Desktop objects

Desktop placement object Layer should be set to Hi5_Plane, such as Box001, set as shown in Figure:



4. Button Settings

For example, Button_Interaction_3, set as shown in Figure:



Related interfaces

1. Hand-related interfaces

Hi5_Interface_Hand script

I. Hand state

```
enum E_Interface_Hand_State
{
       ERelease = -1.
       EPinch = 2.
       ELift = 4.
}
E_Interface_Hand_State GetHandState(out int interactionObjectId)
```

E_Interface_Hand_State returns the hand state, interactionObjectId returns the interaction object Id index

II. Hand gesture recognition status

```
enum Hi5_Glove_Gesture_Recognition_State
       {
              ENone = 0.
              EOk.
              EFist.
        EIndexPoint.
              EHandPlane
       }
Hi5_Glove_Gesture_Recognition_State GetRecognitionState()
```

Hi5_Glove_Gesture_Recognition_State returns the current state of the hand.

2. Hand Event Interface

```
public void MessageFun(string messageKey, object param1, object param2)
        {
            if
(messageKey.CompareTo(Hi5_Glove_Interaction_Message.Hi5_MessageMessageKey.messageHandE
vent) == 0)
            {
                Hi5_Glove_Interaction_Hand_Event_Data data = param1 as
Hi5_Glove_Interaction_Hand_Event_Data;

                switch (data.mEventType)
                {
                                case EEventHandType.EClap:
                                       {
//Clap Event
}
                        break;
                                case EEventHandType.EPoke:
{
//Poke Event
}
                        break;
                                case EEventHandType.EPinch:
                                       {
                                              //Grab Event
}
```

```
                        break;
                    case EEventHandType.EThrow:
{
//Throw Event
}
                        break;
                    case EEventHandType.ELift:
{
//Lift Event
}
                        break;
                    case EEventHandType.ERelease:
{
//Release
}
                        break;
                }
            }
        }
```

3.Interactive Object interface

The state of the interactive object

```
      enum E_Object_State
            {
                    ENone = -1,
                    EStatic = 1,
                    EPinch = 3,
                    EMove = 2,
                    EClap = 4,
                    EFlyLift = 5,
                    EPoke = 6,
 }
 E_Object_State GetObjectItemState();Gets the interactive object state
 int GetObjectId() ;returns the interactive object Id
```

Alternating object event

```
        public void MessageFun(string messageKey, object param1, object param2)
        {
            if
(messageKey.CompareTo(Hi5_Glove_Interaction_Message.Hi5_MessageMessageKey.messageObjec
tEvent) == 0)
            {
                Hi5_Glove_Interaction_Object_Event_Data data = param1 as
Hi5_Glove_Interaction_Object_Event_Data;
                if (data.mObjectId == ObjectItem.idObject)
                {
                    switch (data.mEventType)
                    {
                        case EEventObjectType.EClap:
                            {
                            }
                            break;
                                    case EEventObjectType.EPoke:
                            break;
                        case EEventObjectType.EPinch:
```

```
                            break;
                        case EEventObjectType.EMove:
                            break;
                        case EEventObjectType.ELift:
                            break;
                        case EEventObjectType.EStatic:
                            if (mItem != null)
                            {

                                mItem.ResetCorlor();
                            }
                            break;
                    }
                }
            }
 }
```

4. Button Interface

Hi5_Interface_Button

```
virtual public void MessageFun(string messageKey, object param1, object param2)
        {
            if
(messageKey.CompareTo(Hi5_Glove_Interaction_Message.Hi5_MessageMessageKey.messageObjec
tEvent) == 0)
            {
                Hi5_Glove_Interaction_Object_Event_Data data = param1 as
Hi5_Glove_Interaction_Object_Event_Data;
                if (data.mObjectId == ObjectItem.idObject)
                {
                    if (data.mEventType == EEventObjectType.EClap)
                    {

                    }
                                else if (data.mEventType == EEventObjectType.EPoke)
                    {

                    }
                    else if (data.mEventType == EEventObjectType.EStatic)
                    {
                    }
                }
            }
        }
```