

Hi5 Unity Interaction SDK UserGuide

--V1.1.0.35

Content

Overview.....	2
Setup Environment	3
1. Import SteamVR plugin	3
2. Import the latest version of Hi5_Unity_SDK	3
3. Configure Build Setting	3
4. Configure Unity Layers.....	4
5. Configure Physics Default Contact Offsets	5
6. Import Hi5_Interaction_SDK.untypackage	5
Sample Overview.....	6
1. Sample Scene and Scene Switch	6
Make an object as an Interactive object	7
1. Common Object	7
2. Button Object.....	12
Hi5 Interaction SDK API.....	14
1. Switch Scenes	14
2. API Overview	14
3. Interface API.....	14
4. Message Mechanism (Only for reference)	14
5. Defined Messages	14
6. Handle Defined Messages	14
7. Gesture Recognition Hi5_Interface_Hand	16
Appendix	16
Settings of Hand bounding box.....	16

Overview

Version Control:

Unity Hi5 Interaction SDK: V1.1.0.35

Compatible Hi5 SDK: Hi5_Unity_SDK_1_0_0_655_16

Compatible Unity version: Unity 2017.1.0f3+ (currently not support Unity 2018.3.0)

This document is composed of five parts. Part I: Setup environment; Part II: Sample overview; Part III: Making an object as an interactive object; Part IV: Hi5 interaction SDK API; Part V: Appendix about the configuration of hand bounding box. (It's recommended to use the model delivered with SDK).

The Hi5 Interaction SDK supports the interactions between hands and objects. Currently the functions include grabbing, pinching, releasing, throwing, lifting, poking, hand pressing and exchanging objects.

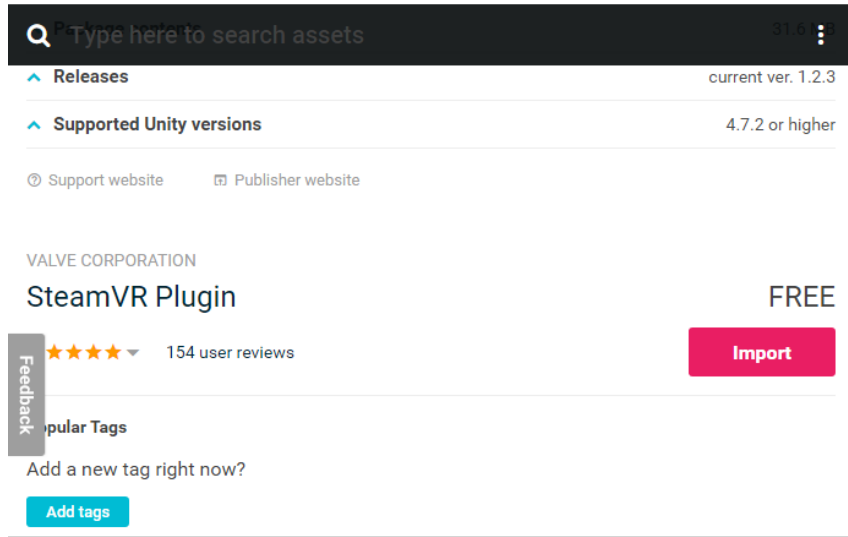
The Hi5 Interaction SDK also supports basic gesture recognitions. Currently it supports 4 gestures: Okay, Flat Palm, Fist and Pointing (index finger) gestures.

More functionality will be added in future versions.

Setup Environment

1. Import SteamVR plugin

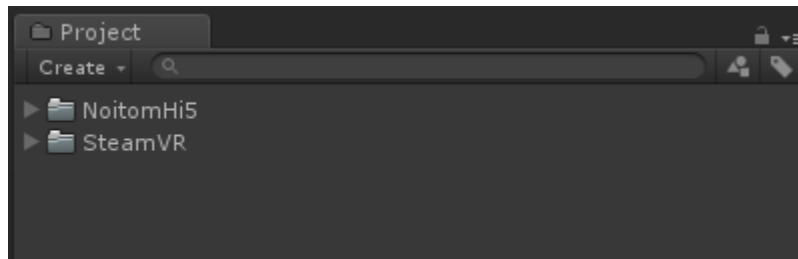
(Please refer to Unity Asset Store.)



2. Import the latest version of Hi5_Unity_SDK

Once completed these 2 steps, the SteamVR and NoitomHi5 will be available.

(The Hi5_Unity_SDK should be version Hi5_Unity_SDK_1_0_0_655_16 or later)



3. Configure Build Setting

Set Target

Select File→Build Setting:

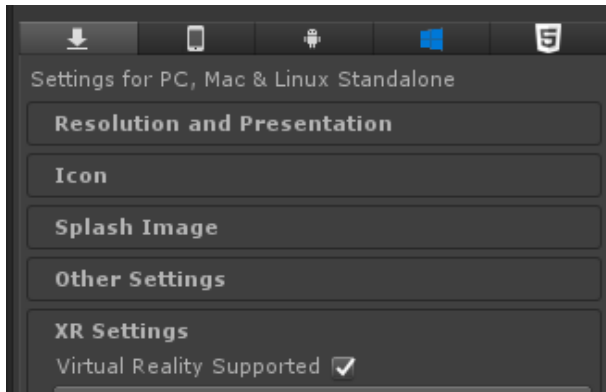
Target Platform: windows

Architecture: x86_64

Set VR setting

Select File→Build Setting→Player Settings:

XR settings: Check Virtual Reality Supported



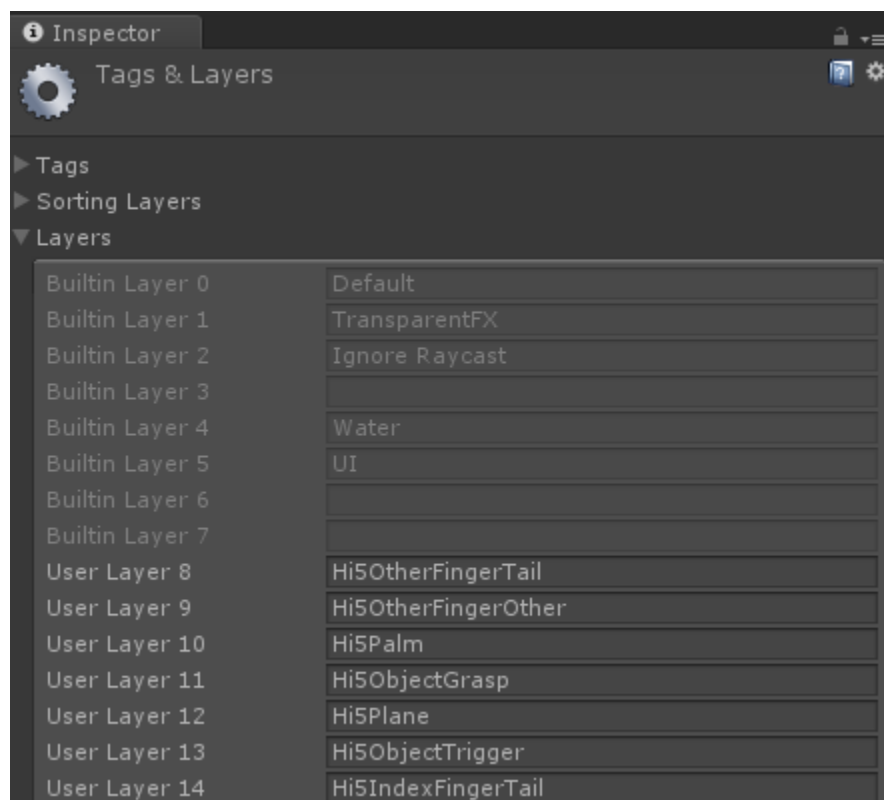
4. Configure Unity Layers

Select Edit→Project Setting→Tags and Layers

Select Layers → Edit layers to set Layers

Set layers as following list, the layer ID is not fixed (the layer order doesn't matter).

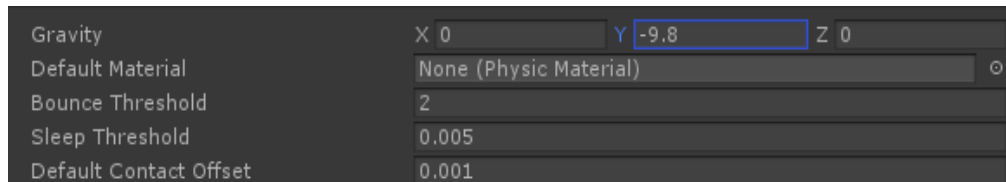
User Layer 8	Hi5OtherFingerTail
User Layer 9	Hi5OtherFingerOther
User Layer 10	Hi5Palm
User Layer 11	Hi5ObjectGrasp
User Layer 12	Hi5Plane
User Layer 13	Hi5ObjectTrigger
User Layer 14	Hi5IndexFingerTail



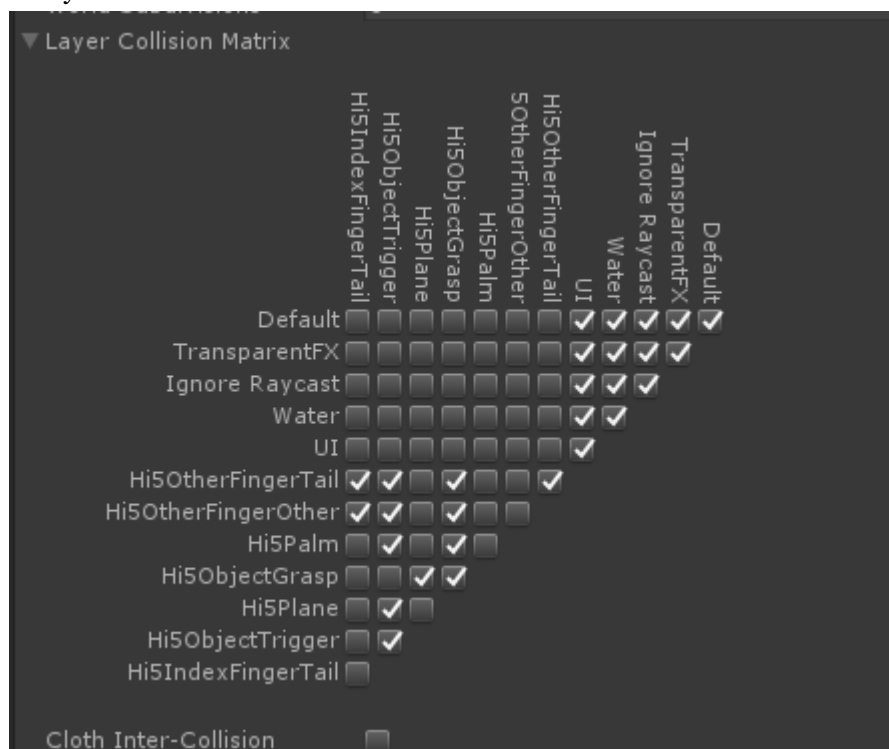
5. Configure Physics Default Contact Offsets

Edit→Project Setting→Physics

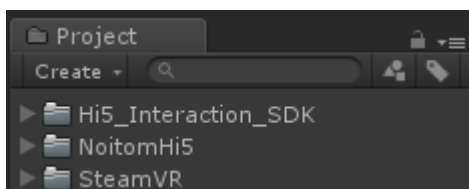
Default Contact Offset: 0.001



Set Layer Collision Matrix as below:



6. Import Hi5_Interaction_SDK.unitypackage



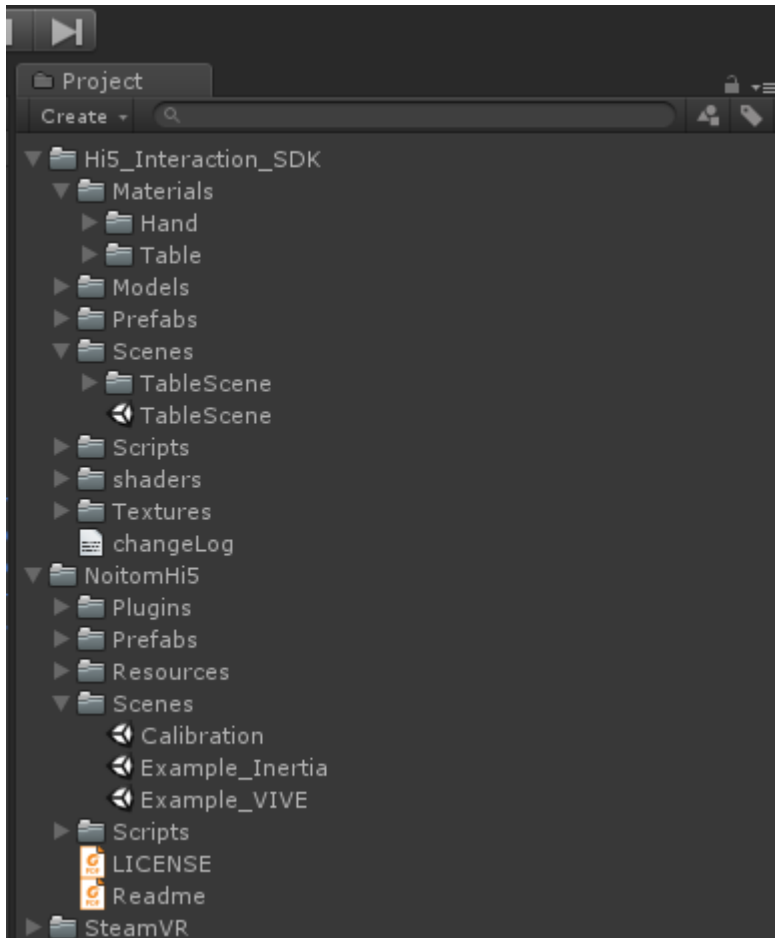
Once imported, the Hi5_Interaction_SDK will be available.

Sample Overview

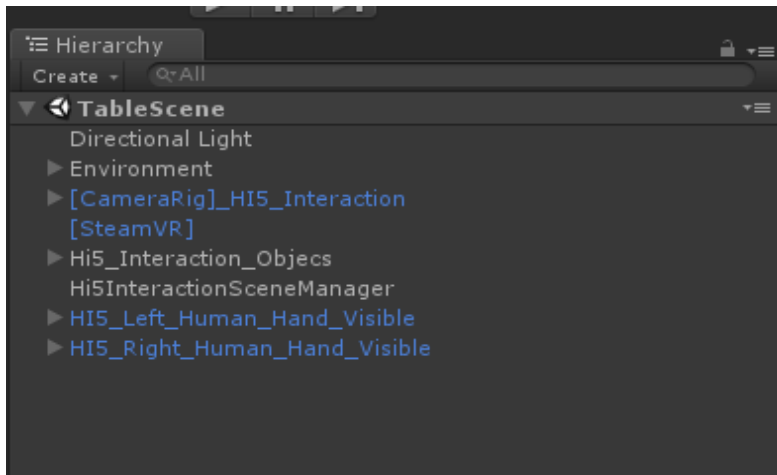
1. Sample Scene and Scene Switch

There are 2 scenes loaded in the sample. They are NoitomHi5/Scenes/Calibration and Hi5_Interaction_SDK/Scenes/TableScene. TableScene is Hi5 interaction SDK example scene. Calibration scene is provided by Hi5 Unity SDK for calibration.

Project View:



TableScene View:



[Hi5InteractionSceneManager](#) script controls the switching of the two scenes. Press “1” to load TableScene and press “2” to load Calibration scene.

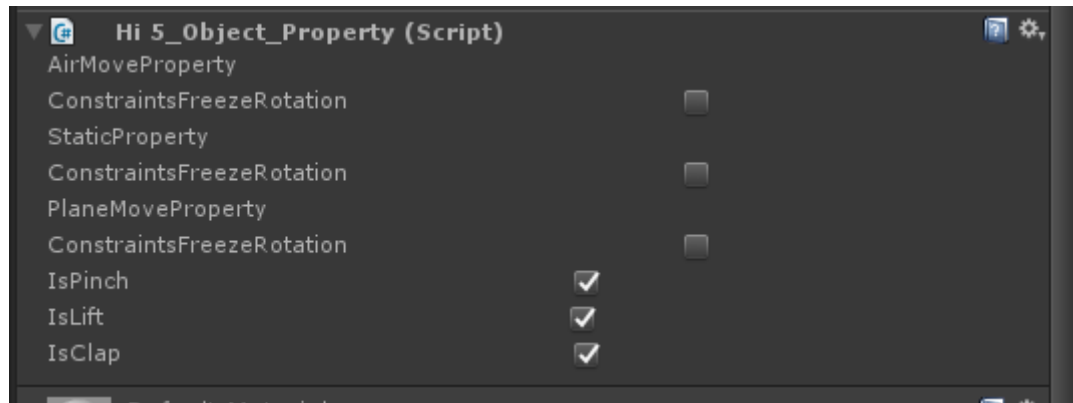
Table Scene

- a. Using prefab [\[CameraRig\]_HI5_Interaction](#) for gloves:
It uses [Hi5InteractionManger](#) script. This script manages the gloves.
There are two models of glove: [HI5_Left_Human_Collider](#) and [HI5_Right_Human_Collider](#). In Interaction SDK, they will not be rendered but use for collision detection and pose recognition.
[HI5_Interaction_Objects](#) include all objects. The script [Hi5_Interaction_Object_Manager](#) manages all these objects for interaction SDK.
The interactive object should be placed under the root note.
- b. [HI5_left_Human_Hand_Visible](#) and [HI5_Right_Human_Hand_Visible](#) will render the left hand and right hand.
- c. Environment includes all other things except interactive objects.

Make an object as an Interactive object

1. Common Object

- a. Add script [Hi5_Glove_Interaction_Item](#)
 - ObjectType: ECommon
 - ID: <IdObject> (unique)
- b. Set Collider Trigger
 - IsTrigger: uncheck
- c. Set Rigidbody
 - Use Gravity: check
 - Is kinematic: uncheck
- d. Add script [Hi5_Interface_Object](#)
- e. Set Layer to “Hi5ObjectGrasp”
- f. Add script [Hi5_Interaction_Item_Collider](#)
- g. Add script [Hi5_Object_Property](#) and Set object interaction Property



h.

Air Friction Rate: air resistance of this object

Air Move Property: The rotation constraint while moving through the air

Static Property: The rotation constraint while static

Plane Move Property: The rotation constraint while moving on the surface of plane

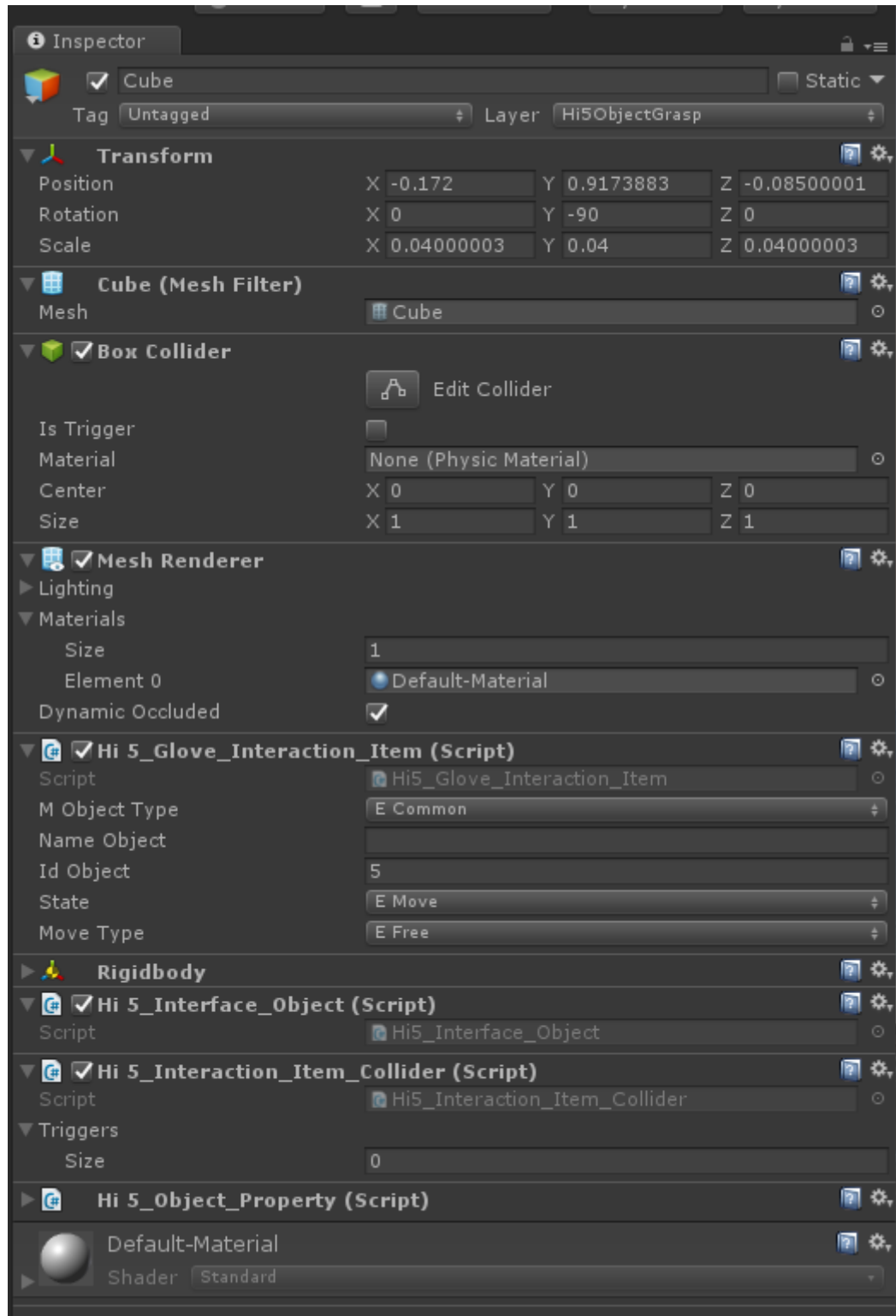
IsPinch: enable/disable the capable to be pinched of this object

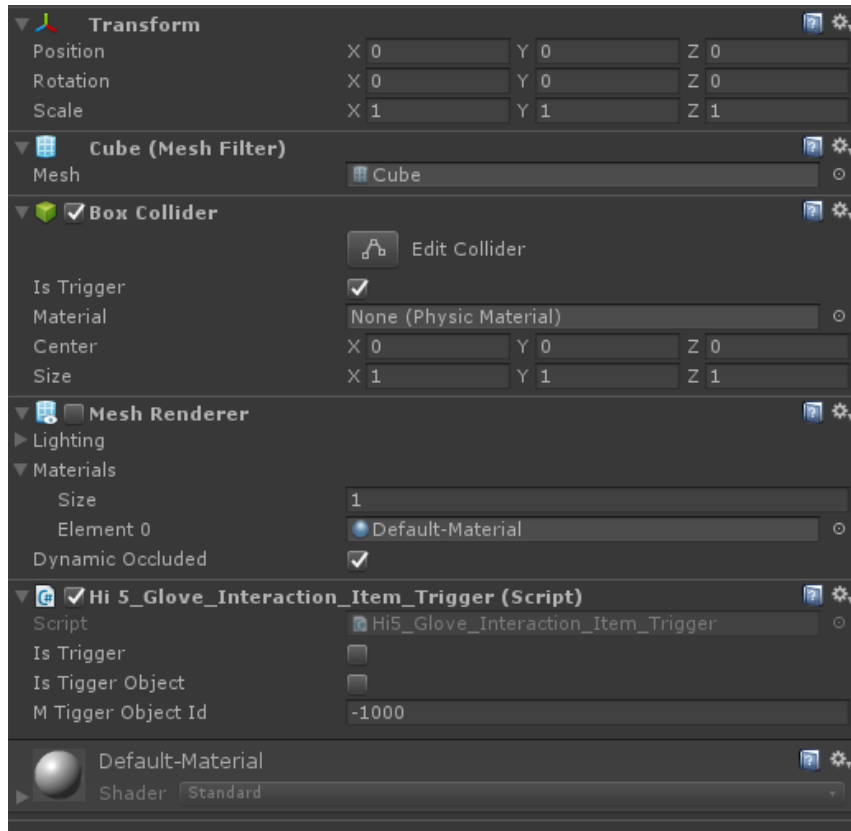
IsClap: enable/disable the capable to be clapped of this object

IsLift: enable/disable the capable to be lifted of this object

i. Add sub-object

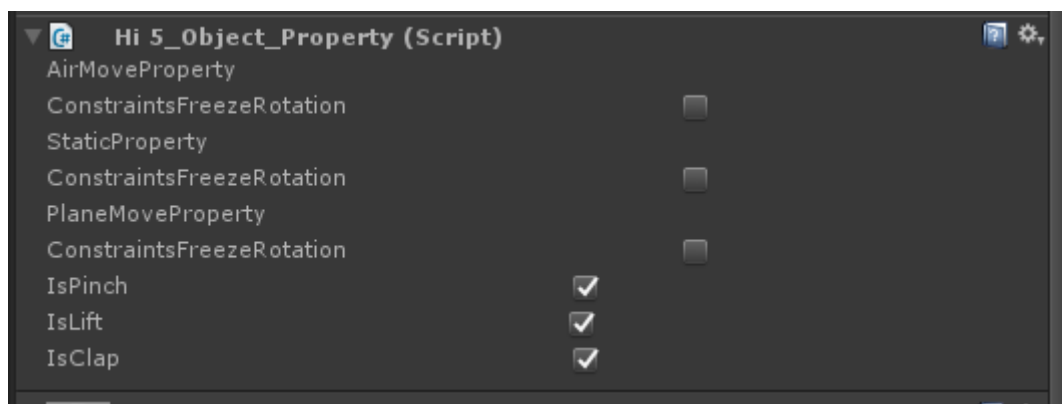
- Sub-object uses the same model as parent object and disable render.
- Set Layer to "Hi5ObjectTrigger"
- Collider IsTrigger: Check.
- Add script [Hi5_Glove_Interaction_Item_Trigger](#)





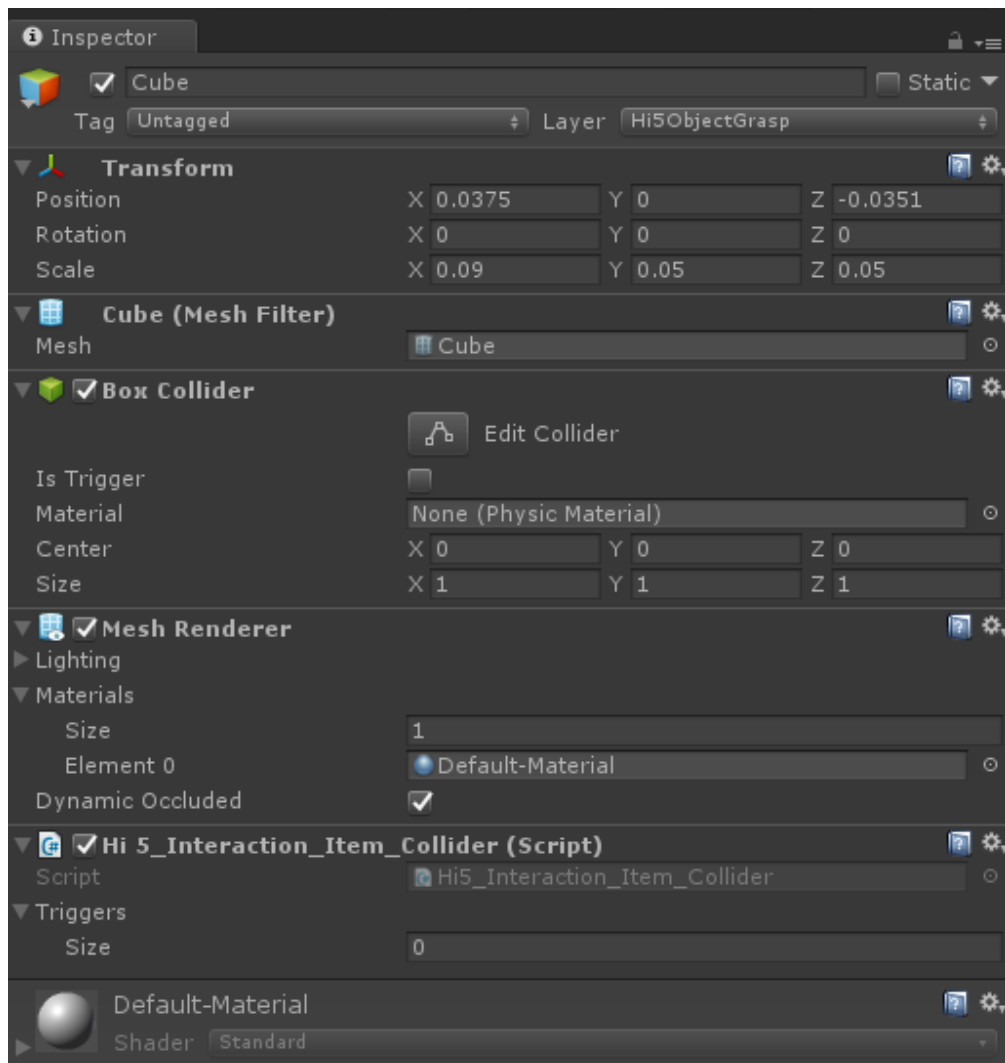
Object Collider assembly

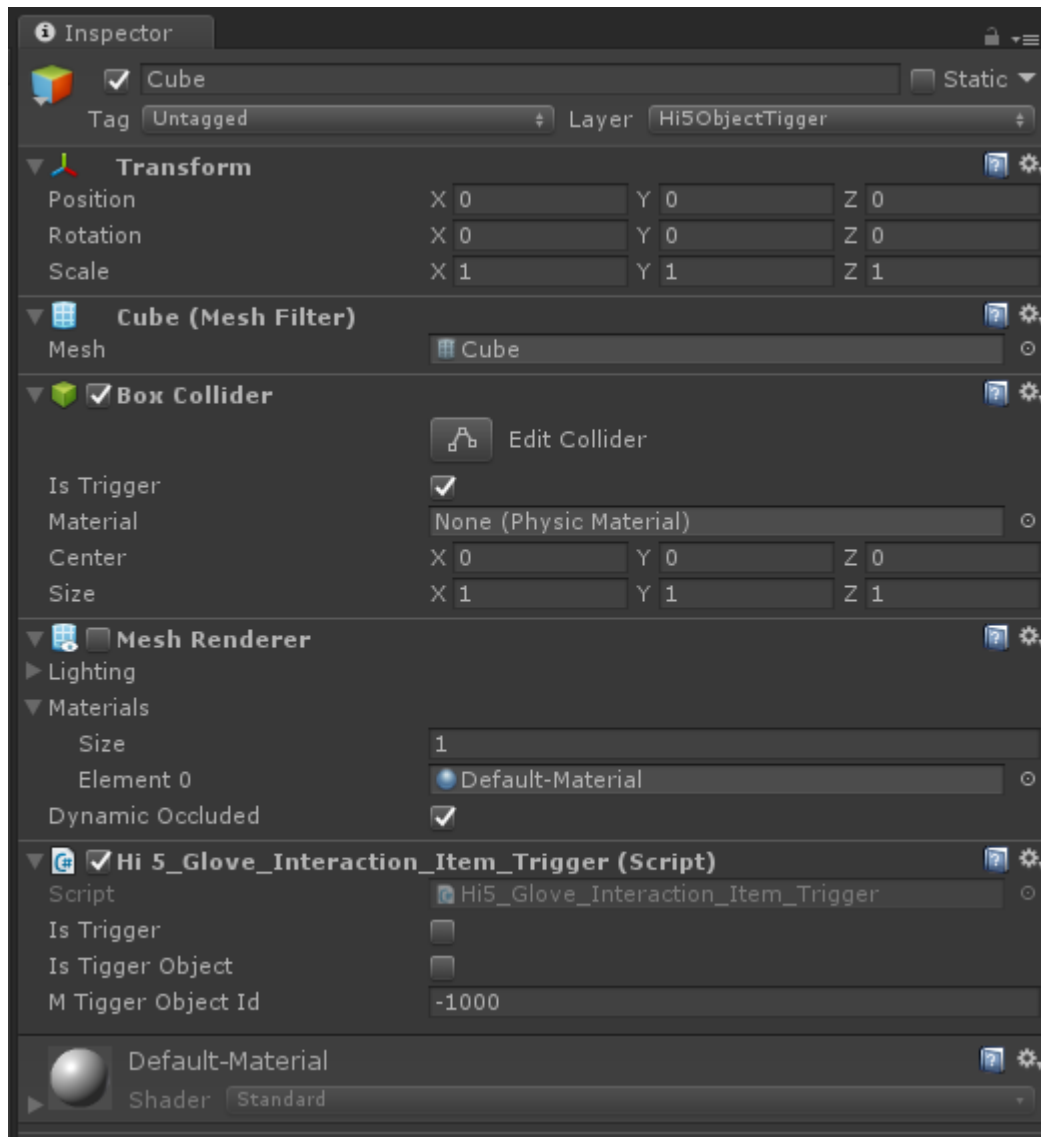
- Add script [Hi5_Glove_Interaction_Item](#)
 - ObjectType: ECommon
 - ID: <IdObject> (unique)
- Set Rigidbody
 - Use Gravity: check
 - Is kinematic: uncheck
- Add Hi5_Interface_Object script and Hi5_Object_Property script.
Set object property:



- Set assembled object.
 - Add script Hi5_Interaction_Item_Collider
 - Set Layer to "Hi5ObjectGrasp"

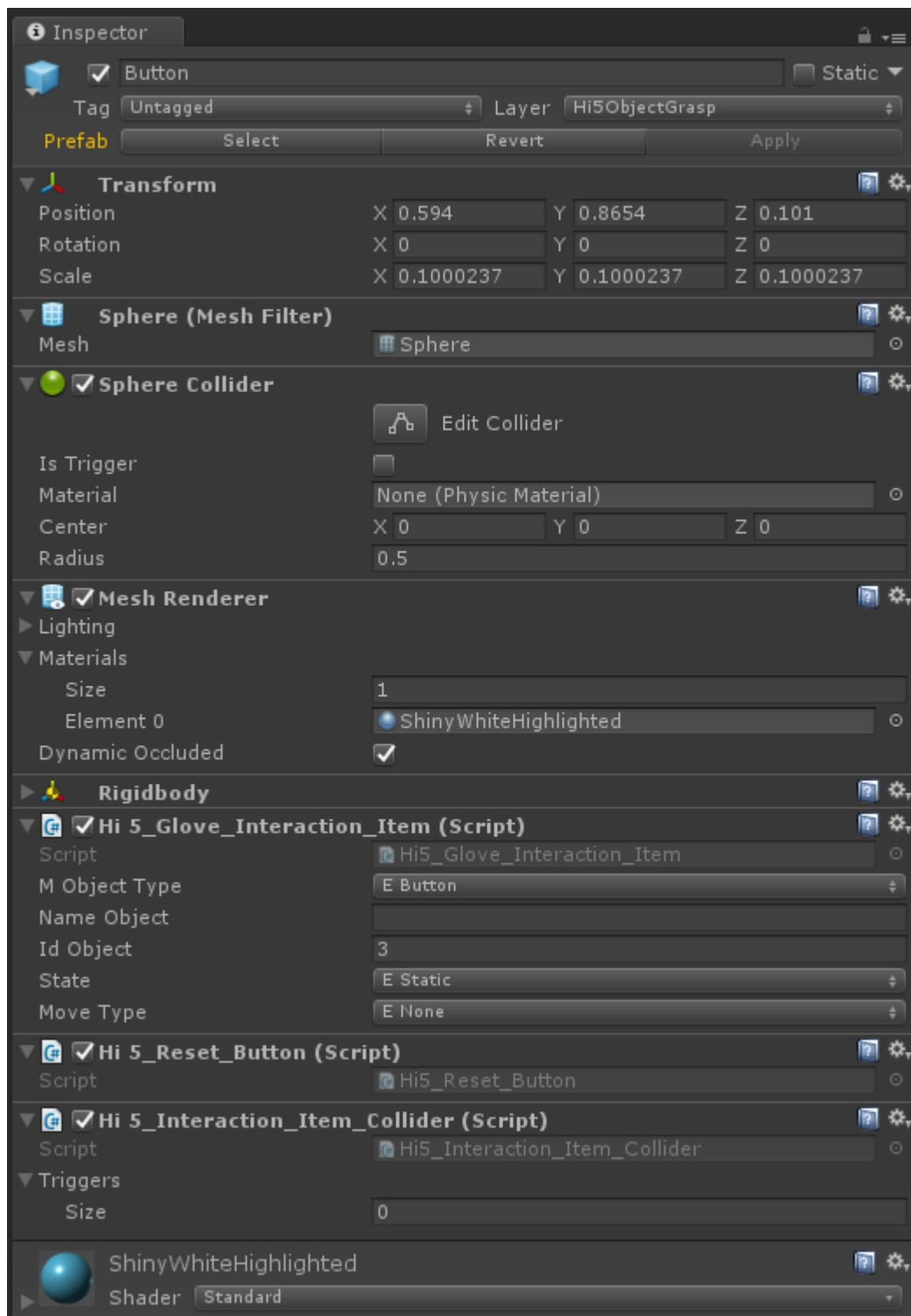
- Collider IsTrigger: Uncheck
- Add sub-object
 - Set Layer to “Hi5ObjectTrigger”
 - Collider IsTrigger: Check
 - Add script [Hi5_Glove_Interaction_Item_Trigger](#)
- e. Set Cubelift object
 - . Set Layer to “Hi5ObjectGrasp”
 - .Collider IsTrigger: Check
 - .Add scripe [Hi5_Object_Lift_Collider](#)





2. Button Object

- a. Set Layer to “Hi5ObjectGrasp”
- b. Add script [Hi5_Glove_Interaction_Item](#)
 - ObjectType: EButton
 - ID: <IdObject> (unique)
- c. Set Trigger
 - IsTrigger: uncheck
- d. Set Rigidbody
 - Use Gravity: check
 - Is kinematic: check
 - Constraints→Freeze Position x y z: check all
 - Constraints→Freeze Rotation x y z: check all
- e. Add script [Hi5_Interface_Button](#)
- f. Add script [Hi5_Interaction_Item_Collider](#)



Hi5 Interaction SDK API

1. Switch Scenes

(Please refer to Hi5InteractionSceneManager.cs)

2. API Overview

Interaction SDK has two parts of script, Core and Interface. **Core** wraps many basic functions and physics implementations. Developers could use **interface** to get interaction messages and handle event.

3. Interface API

- a) Hand
 - i. `Class Hi5_Interface_Hand`
- b) Object
 - i. `Class Hi5_Interface_Object`
 - ii. `Class Hi5_Interface_Button`

4. Message Mechanism (Only for reference)

Message/Hi5_Glove_Interaction_Message.cs

Message/Hi5_Interaction_Message.cs

- a) `public static Hi5_Interaction_Message GetInstance()`
- b) `public void RegisterMessage(MessageFun fun, string messageKey)`
- c) `public void UnRegisterMessage(MessageFun fun, string messageKey)`
- d) `public void DispenseMessage(string messageKey, object param1, object param2 = null, object param3 = null, object param4 = null)`

5. Defined Messages

Interaction Message: All that should be handled by Developers are these two kinds of messages.

- Condition: When interaction status changes (start/stop interaction)
- Messages: Both two kinds of messages will be sent
- Message Type:
 - 1. `messageObjectEvent`
 - 2. `messageHandEvent`

6. Handle Defined Messages

In each Interface Class, there is a MessageFun function to handle corresponding messages.

MessageFun will receive the interaction messages that send by SDK core, and

developer can handle them here.

a) For Hand: Hi5_Interface_Hand (Interface/Hi5_Interface_Hand.cs)

```
void MessageFun(string messageKey, object param1, object param2)
{
    Hi5_Glove_Interaction_Hand_Event_Data data = param1 as
    Hi5_Glove_Interaction_Hand_Event_Data;
}
```

● Related Data Structure

```
public class Hi5_Glove_Interaction_Hand_Event_Data
{
    public int mObjectId = -1;    // the ID of interaction object
    public EEventHandType mEventType = EEventHandType.ENone;
    public EHandType mHandType = EHandType.EHandLeft;
}

public enum EEventHandType
{
    ENone = 0,
    EClap,
    EPoke,
    EPinch,
    EThrow,    //previous status is EPinch
    ELift,
    ERelease,    //previous status if ELift/EClap
}
```

b) For Common Object: Hi5_Interface_Object

```
void MessageFun(string messageKey, object param1, object param2){}
Hi5_Glove_Interaction_Object_Event_Data data;
```

● Related Data Structure

```
public class Hi5_Glove_Interaction_Object_Event_Data
{
    public int mObjectId = -1;
    public EObject_Type mObjectType = EObject_Type.ECommon;
    public EHandType mHandType = EHandType.EHandLeft;
    public EEventObjectType mEventType = EEventObjectType.ENone;
}

public enum EEventObjectType
{
    ENone = 0,
    EClap,
    EPoke,
```

```

        EPinch,
        EMove,
        ELift,
        EStatic    //still
    }

```

c) Hi5_Interface_Button

```

void MessageFun(string messageKey, object param1, object param2){}
Hi5_Glove_Interaction_Object_Event_Data data;
● Related Data Structure
public class Hi5_Glove_Interaction_Object_Event_Data
{
    public int mObjectId = -1;
    public EObject_Type mObjectType = EObject_Type.ECommon;
    public EHandType mHandType = EHandType.EHandLeft;
    public EEventObjectType mEventType = EEventObjectType.ENone;
}

```

7. Gesture Recognition Hi5_Interface_Hand

```

public Hi5_Glove_Gesture_Recognition_State GetRecognitionState()
{
    return mHand.GetHand().GetRecognitionState();
}

```

Get hand gesture recognition status.

```

public enum Hi5_Glove_Gesture_Recognition_State
{
    ENone = 0,
    EOk,
    EFist,
    EIndexPoint,
    EHandPlane
}

```

Appendix

Settings of Hand bounding box:

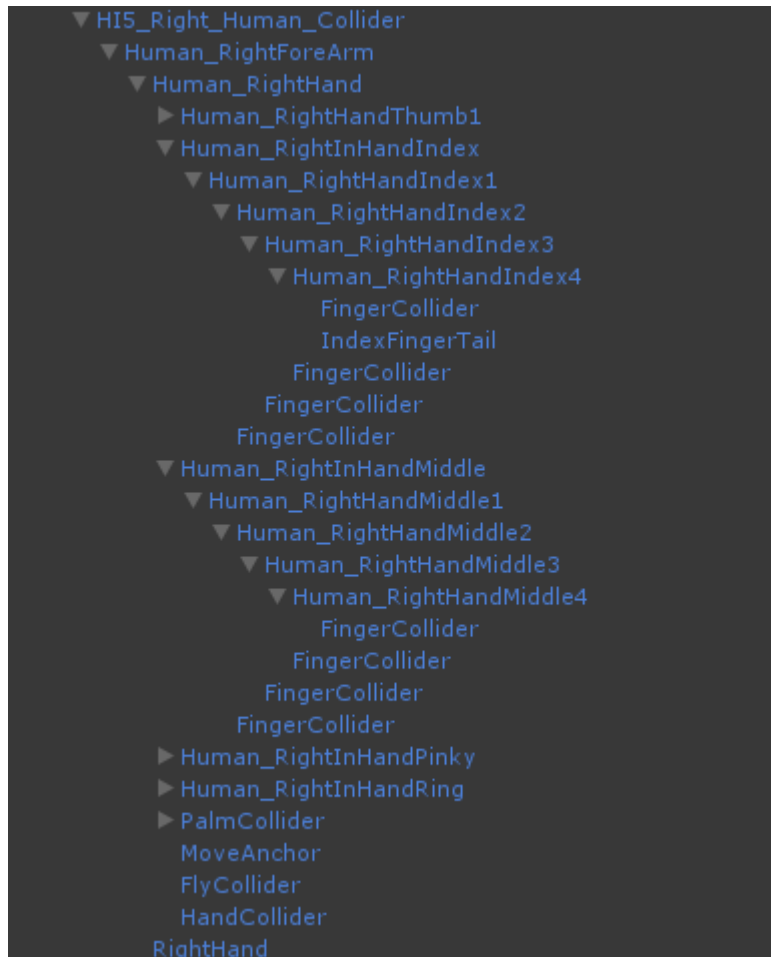
(***Recommend to use SDK default hand model, and please DO NOT modify the name of any finger.)

This project is using two same hand model for one hand. The original SDK hand model is for Gesture Recognition and collision detection; another model is for render. Both two

models have bounding box.

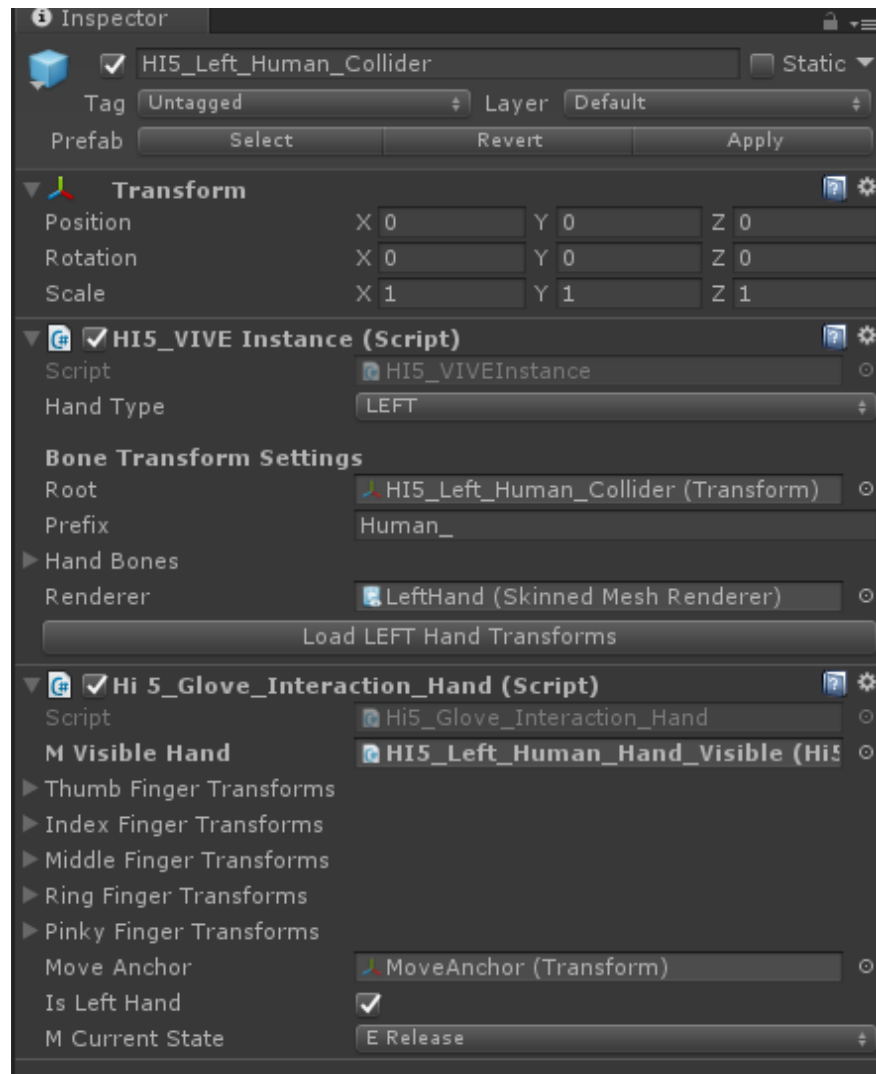
HI5_[Left/Right]_Human_Collider:

1. Hand Skeleton



2. Hand Root node (HI5_Left/Right_Human_Collider) setting

- a) The root node use [Hi5_Glove_Interaction_Hand](#) script
- b) Set Parameter:
 - i. Is_Left_Hand //set left/right hand
 - ii. mVisibleHand: HI5_Left/Right_Human_Hand_Visible prefab //set by drag



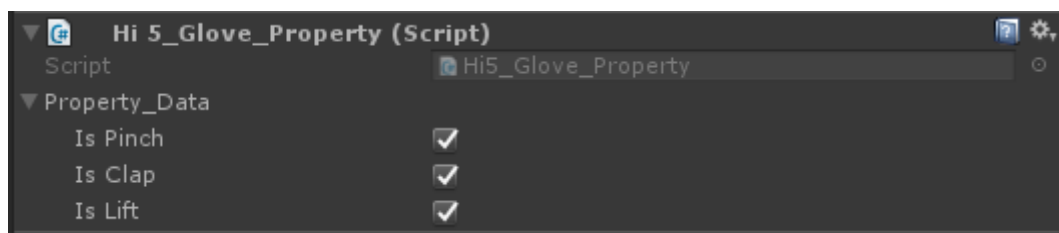
c) [Hi5_Glove_Property](#)

Provide the interface to control the interaction between hand and objects:

IsPinch: enable/disable pinch for this hand

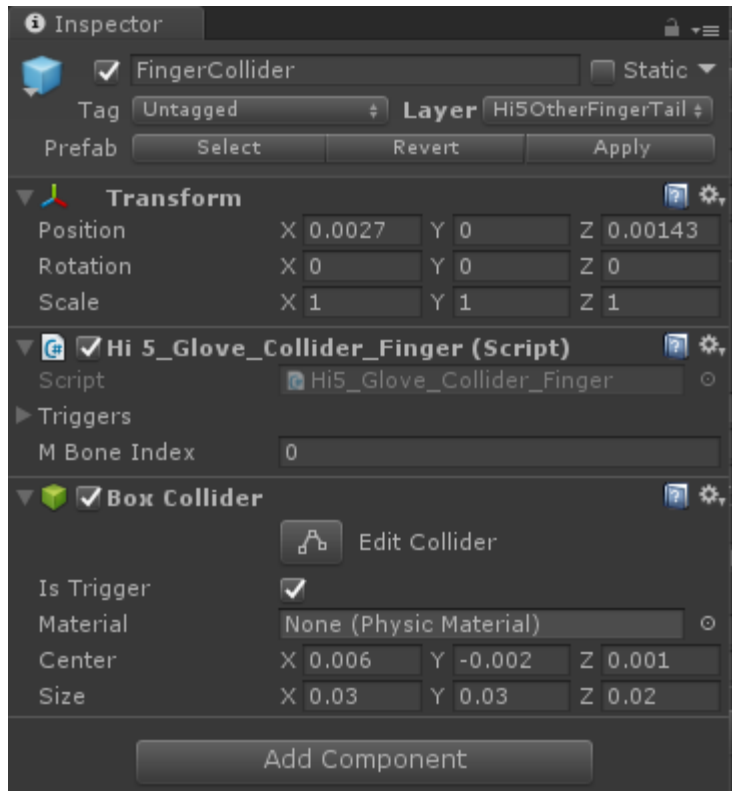
IsClap: enable/disable clap for this hand

IsLift: enable/disable lift for this hand

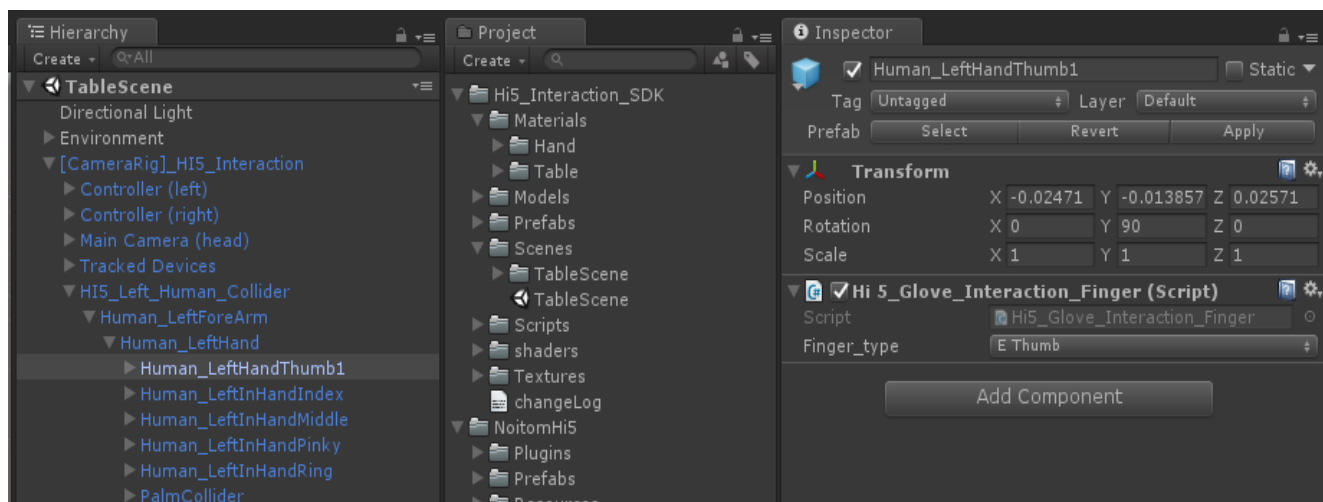


3. Finger node (FingerCollider) setting

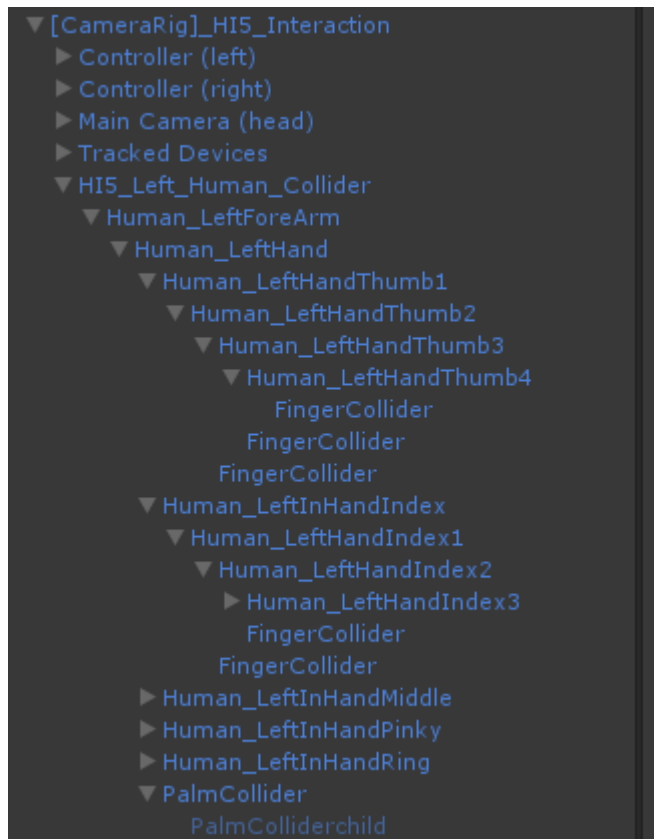
a) Use [Hi5_Glove_Collider_Finger](#) script



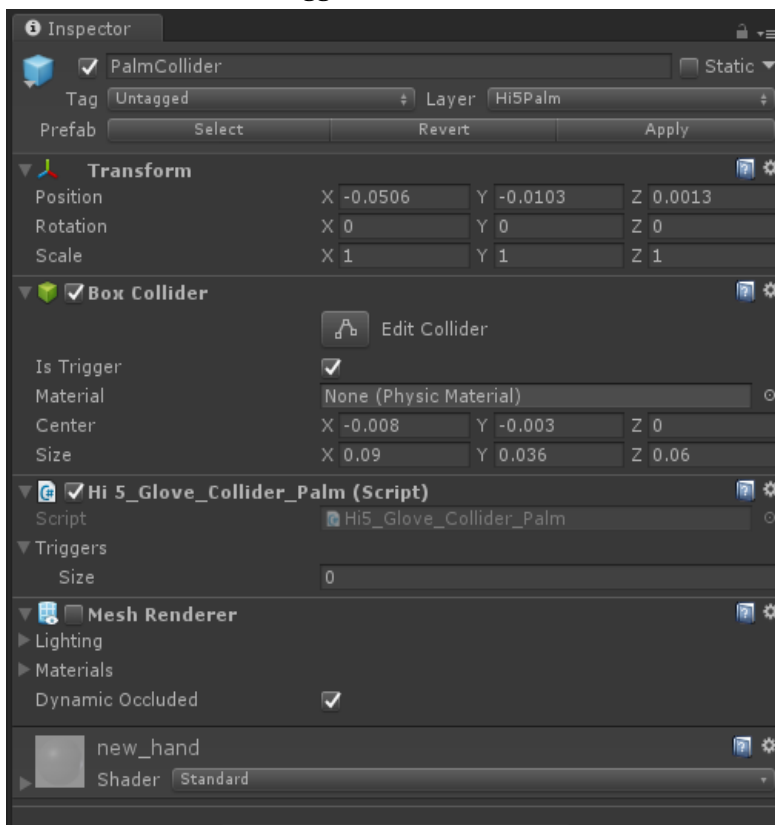
- b) Set Layers
 - i. Finger4: Layer Hi5OtherFingerTail
 - ii. Finger except: Layer Hi5OtherFingerOther
- c) Box Collider → Is Trigger: checked
- 4. Finger root node setting
 - a) Use [Hi5_Glove_Interaction_Finger](#) script



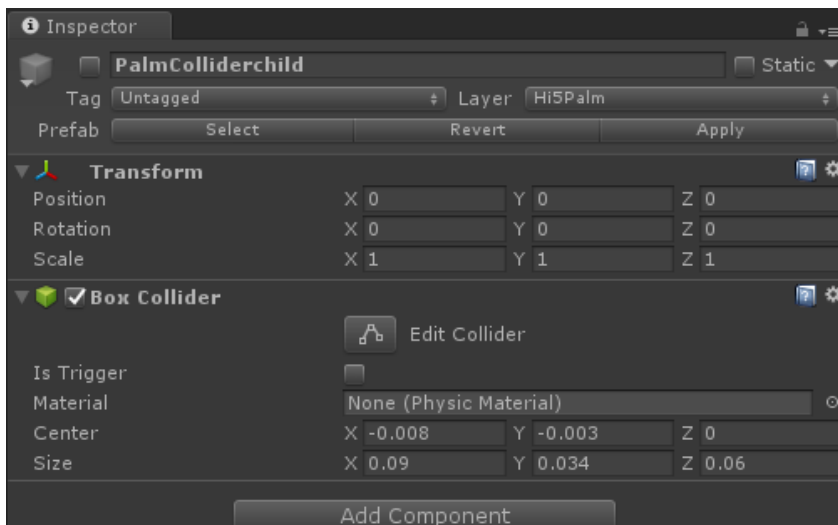
- b) Finger type: E Thumb/E Index/...
- 5. Palm bounding box
 - a) Hierarchy



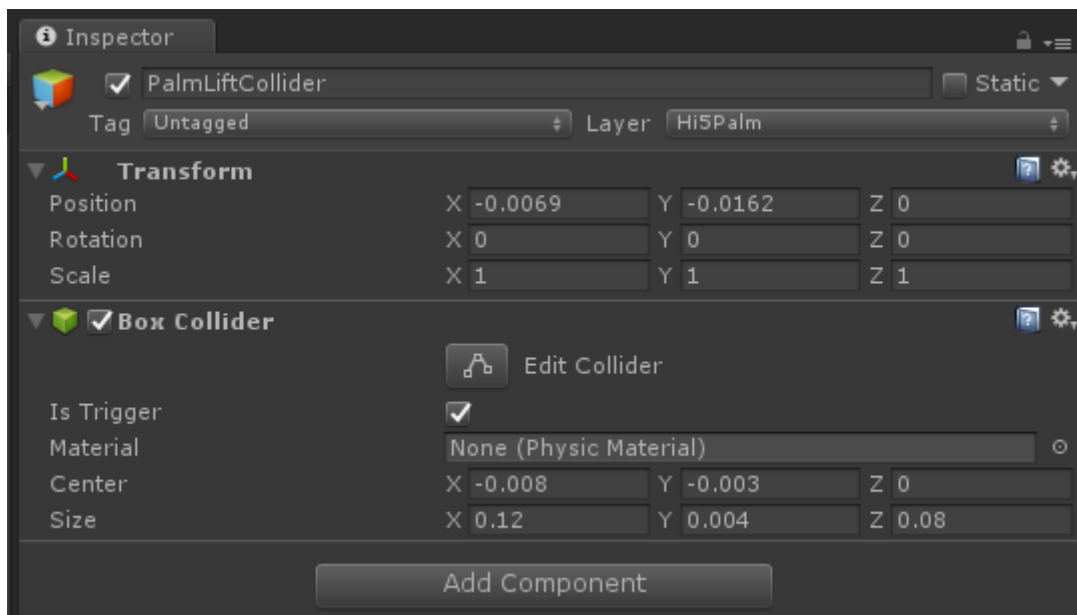
- b) Use [Hi5_Glove_Collider_Palm](#) script
- c) Set Layer: Layer Hi5Palm
- d) Box Collider→IsTrigger: checked



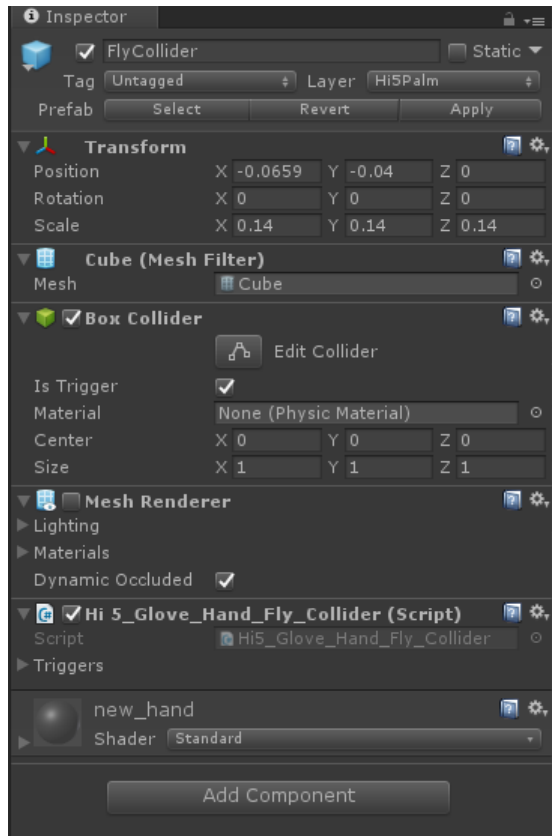
- e) Create sub object: PalmColliderchild
 - i. Set Layer: Layer Hi5Palm
 - ii. Box Collider→IsTrigger: unchecked
 - iii. default: disable



- f) create sub object : PalmLiftCollider or PalmrightCollider
 - i. Set Layer: Layer Hi5Palm
 - ii. Box Collider→IsTrigger: checked



1. FlyCollider

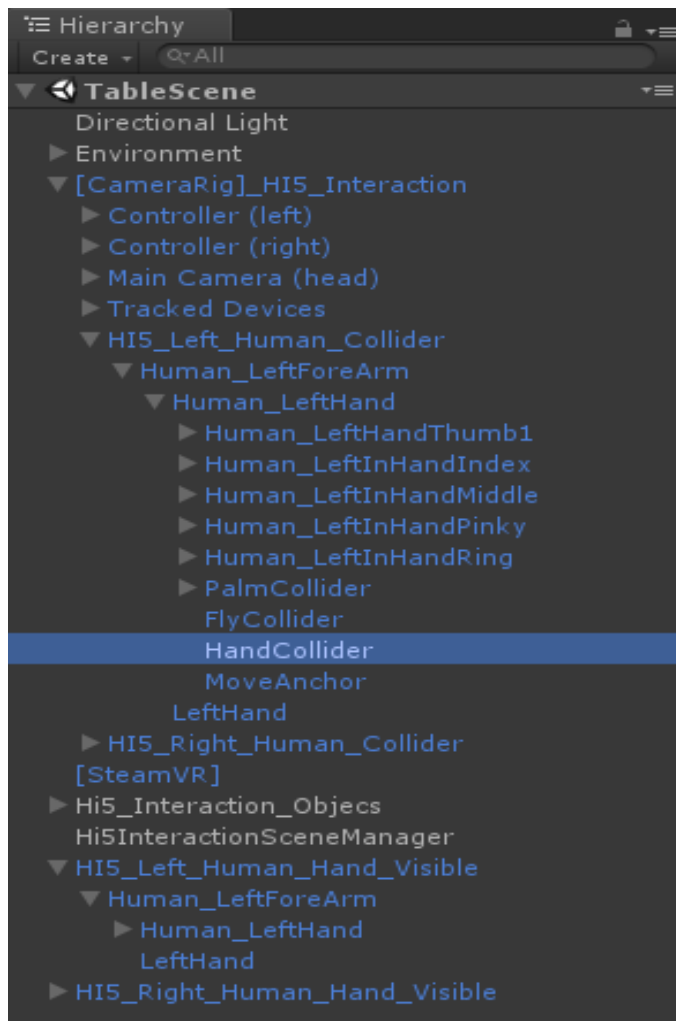


2.

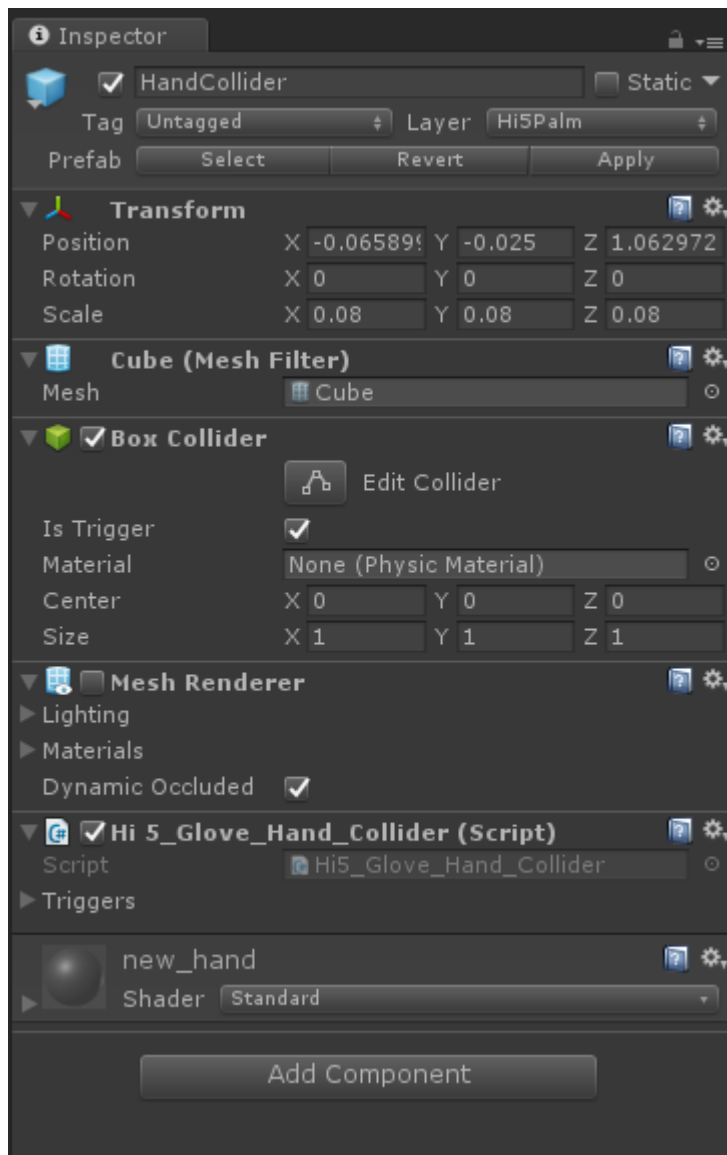
- a) Use [Hi5_Glove_Hand_Fly_Collider](#) script
- b) Set Layer: Hi5Palm
- c) Box Collider → IsTrigger: checked

3. HandCollider

- a) Hierarchy



- b) Use [Hi5_Glove_Hand_Collider](#) script
- c) Set Layer: Hi5Palm
- d) Box Collider→IsTrigger: checked



10 Add `Hi5_Glove_Property` script.

The main function is to set whether gloves could pinch .clap or lift objects.

```
public bool IsPinch
{
    set { Property_Data.IsPinch = value; }
    get { return Property_Data.IsPinch; }
}

public bool IsClap
{
    set { Property_Data.IsClap = value; }
    get { return Property_Data.IsClap; }
}
```



```

public bool IsLift
{
    set { Property_Data.IsLift = value; }
    get { return Property_Data.IsLift; }
}

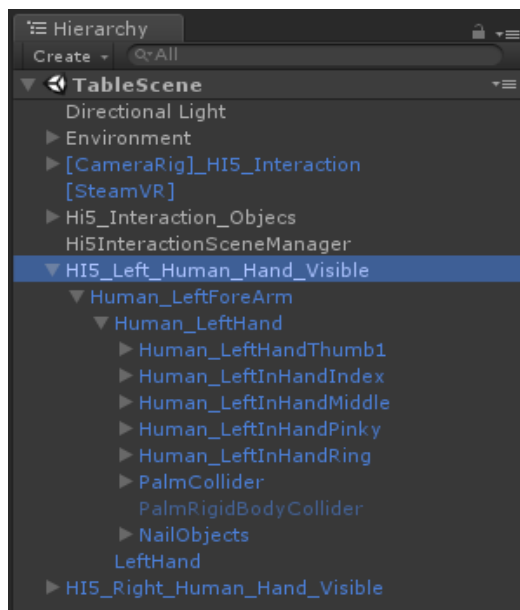
```

11 Add IndexFingerTail

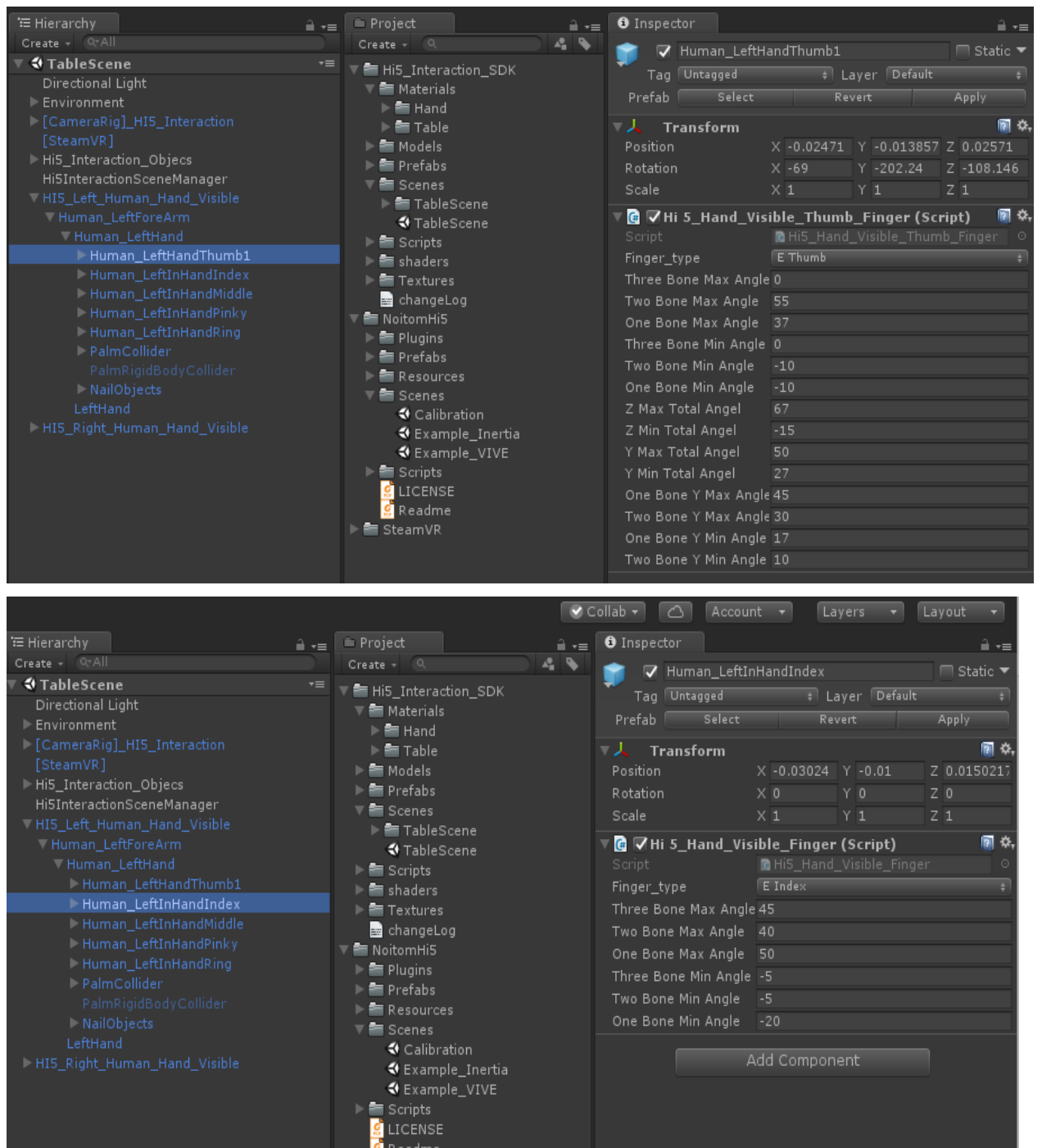
The main fuction is to judge the Ok gesture.

HI5_Left/Right_Human_Hand_Visible

1. Hand Skeleton

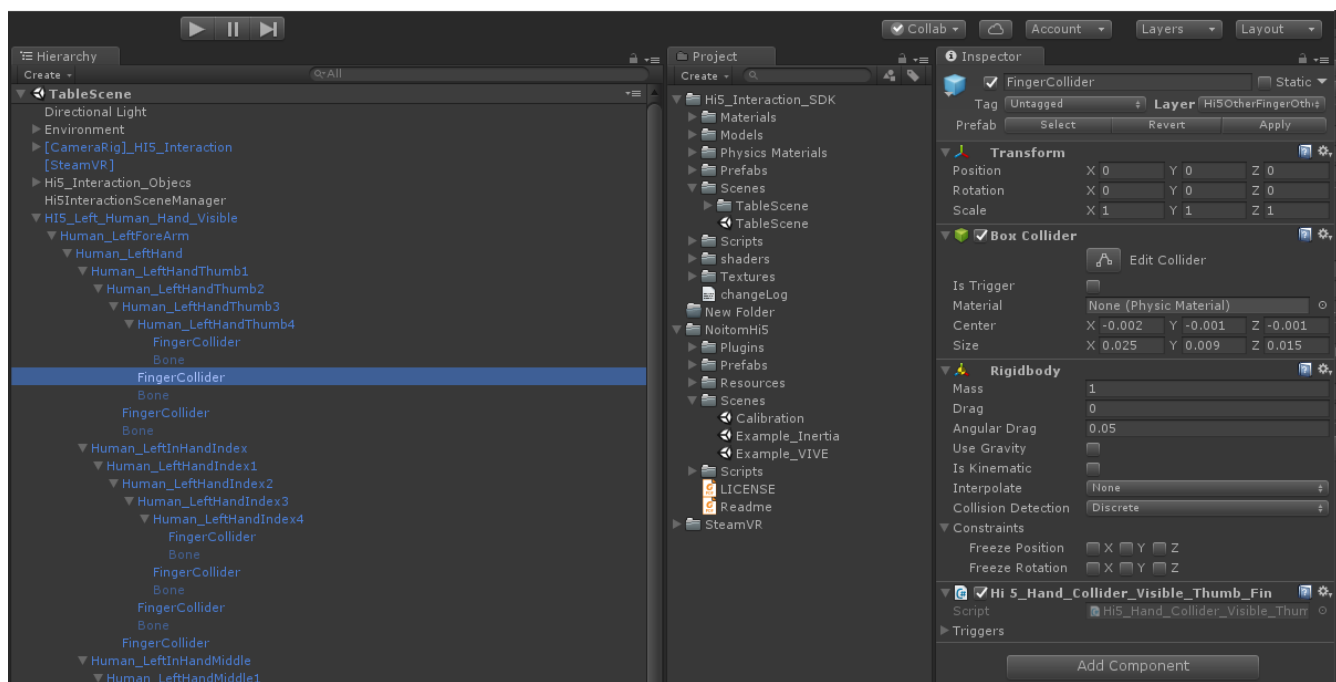
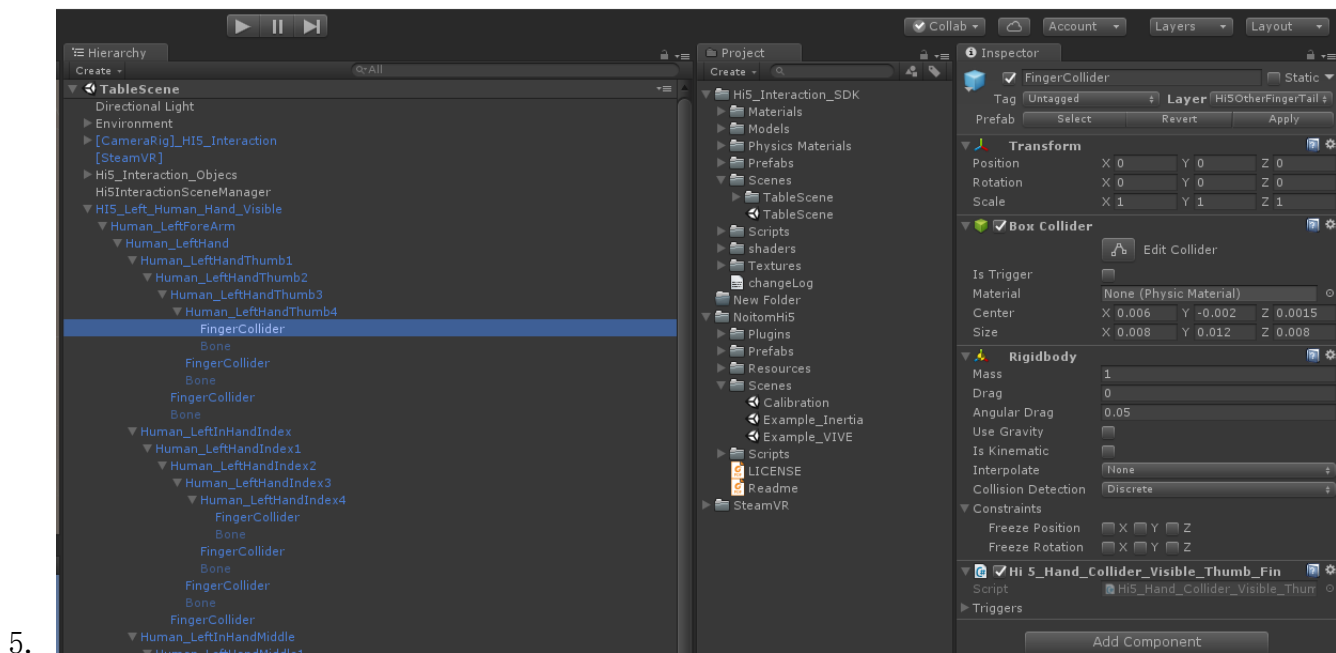


2. Hand Root node (HI5_Left/Right_Human_Hand_Visible) setting
 - a) Use [Hi5_Hand_Visible_Hand](#) script
 - i. MGlove_Hand: HI5_Left/Right_Human_Collider //set by drag
 - ii. ArmTransform: Human_Left/RightForeArm //set by drag
 - iii. HandTransform: Human_Left/RightHand //set by drag
 - iv. Is Left Hand: //Hand type
 - b) Use [Hi5_Interface_Hand](#) script
3. Finger Root node setting
 - a) Use [Hi5_Hand_Visible_Thumb_Finger](#) script for Thumb; Use [Hi5_Hand_Visible_Finger](#) script for other fingers
 - i. Finger_Type: E Thumb/E Index/...
 - ii. Set the maximum bending of each finger



4. FingerCollider

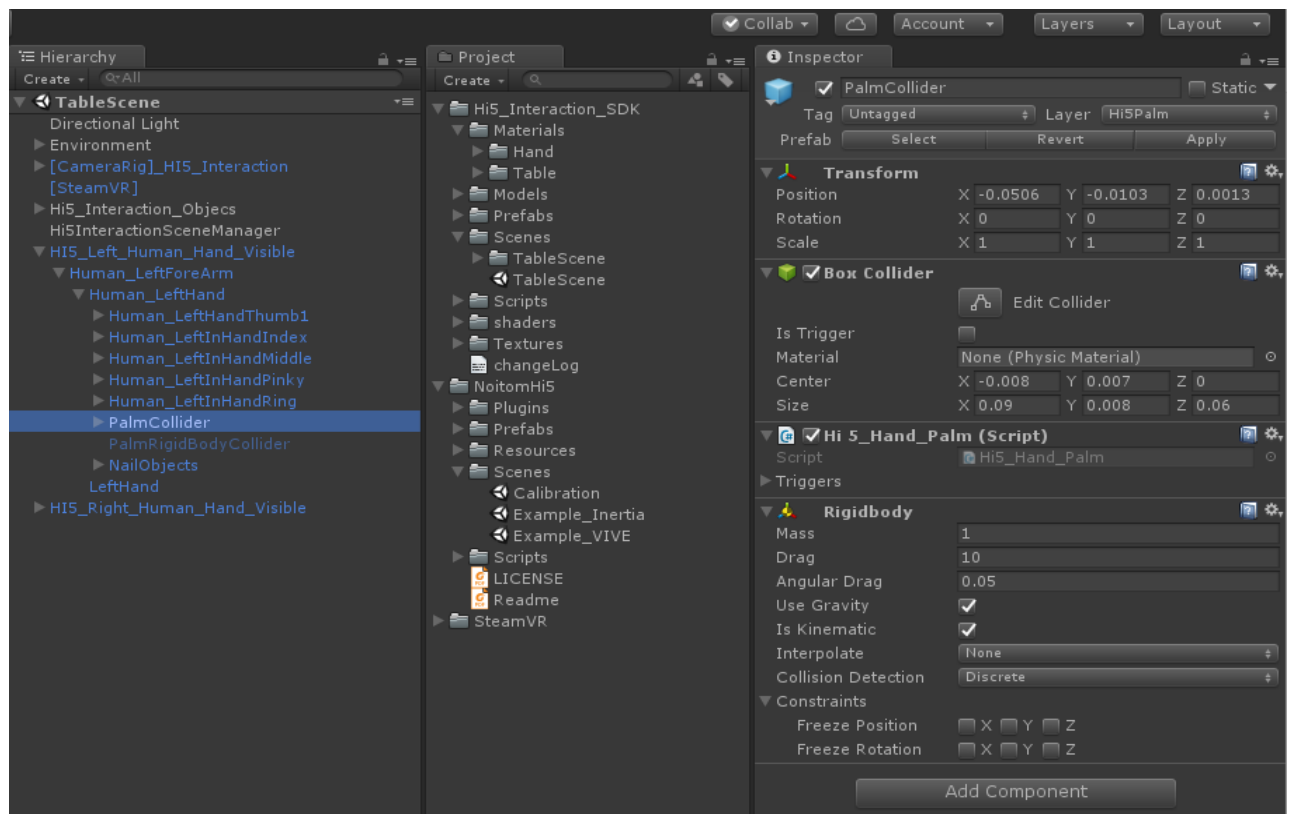
- Use [Hi5_Hand_Collider_Visible_Thumb_Finger](#) script for thumb;
Use [Hi5_Hand_Collider_Visible_Finger](#) script for other fingers.



- Set Layers
 - Finger4: Layer Hi5OtherFingerTail
 - Finger3: Layer Hi5OtherFingerOther
- Box Collider → IsTrigger: uncheck
- Add component: Rigidbody
 - Use Gravity: uncheck
 - Is Kinematic: uncheck

6. PalmCollider

- a) Use [Hi5_Hand_Palm](#) script
- b) Set Layer: Hi5Palm
- c) Box Collider→IsTrigger: uncheck
- d) Add component: Rigidbody
 - i. Use Gravity: checked
 - ii. Is Kinematic: checked



7. NailObject (to hold object while pinch)

- i. Set Layer: Hi5Palm
- ii. Box Collider→IsTrigger: checked
- iii. Use [Hi5_Hand_Nail_Collider](#) script

