NJIT

HCInvestigationInterface

October 25, 2018

1 Phys HCInvestigator

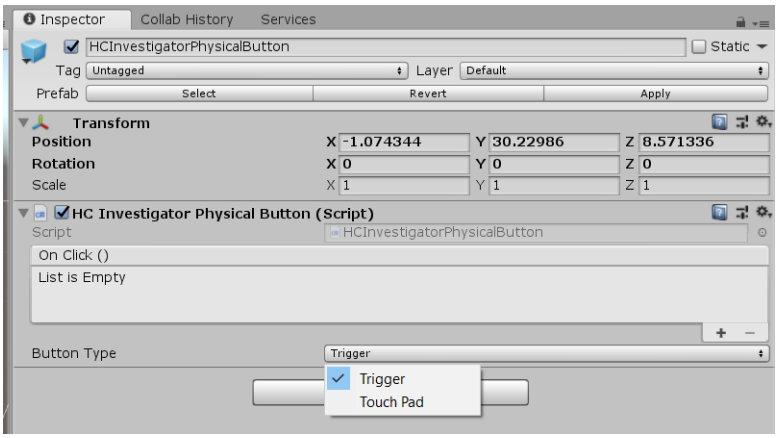
This section goes over implementing the Phys HCInvestigator into a unity project.

1.1 Adding the Prefab

The first step to getting the Phys Investigator to work is adding the prefab into the scene this is done by finding the prefab in the Prefabs folder and selecting the HCInvestigatorPhysicalButton prefab and dragging it into the scene.

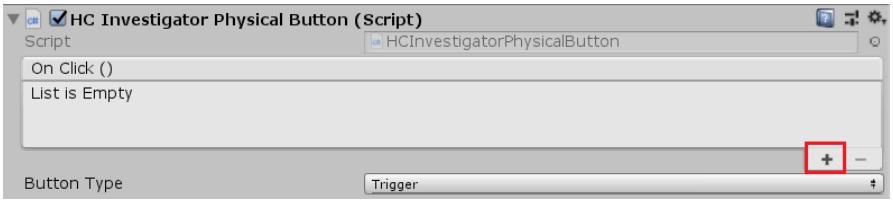
1.2 Selecting the Button

The second step in setting up the Phys HCInvestigator is selecting which button you want the user to interact with on the controller. This can be chosen by dropping down the Button Type variable and using the other drop down option for the Button variable to select which button you want.

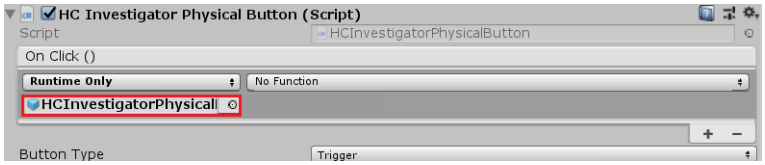


1.3 Selecting the Type of Recording

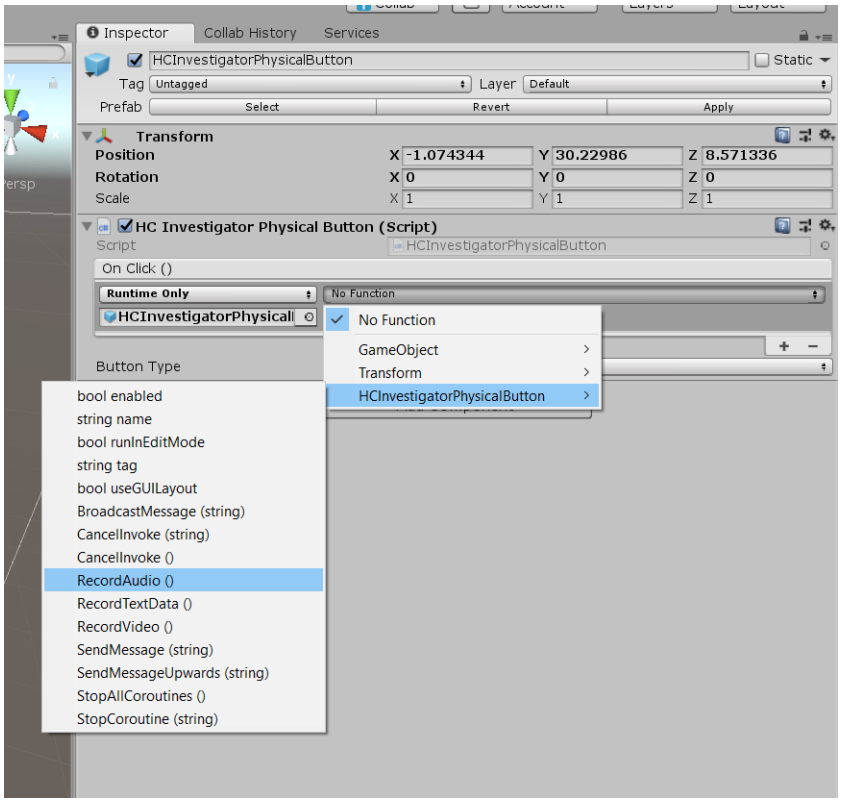
The last step for setting up the Phys HCInvestigator is choosing which type of data you want to record. This takes a number of steps; first is hitting the plus button in OnClick().



The second step is populating the object box with the prefab itself this can be done either by dragging the prefab into the box or searching for it by pressing the little circle next to the box.



The last step is adding which type of recording you want, this is done by navigating the functions tab and hovering over the HCInvestigatorPhysicalButton tab and then selecting which function you want to active.

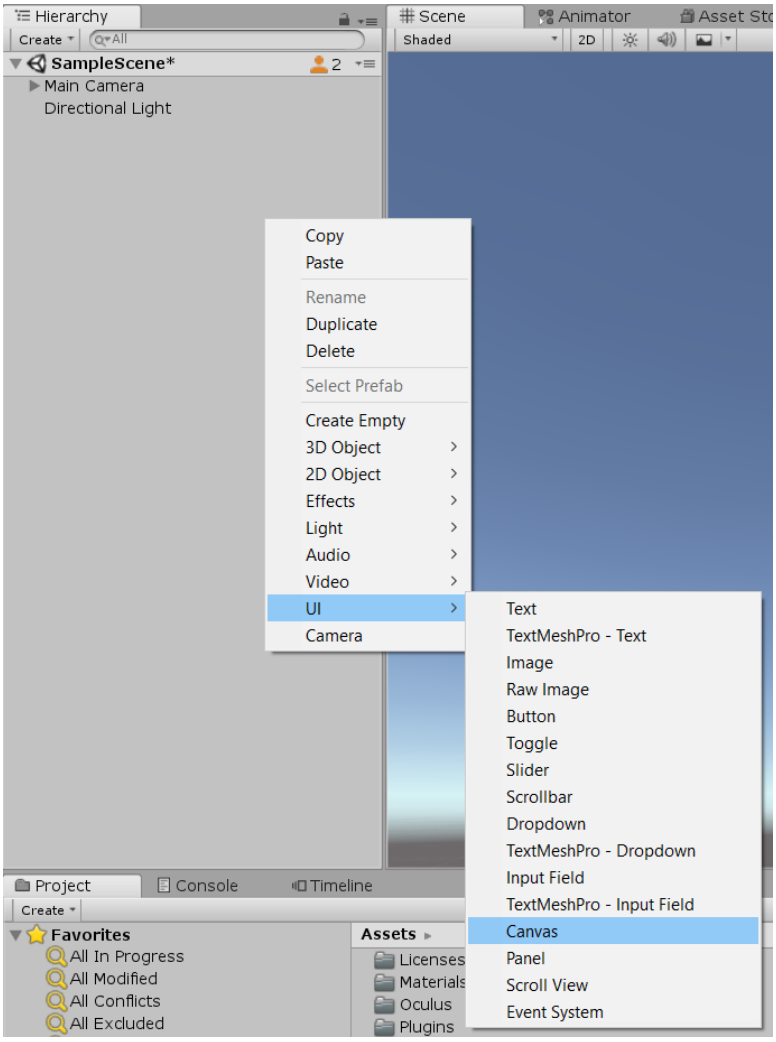

Inspector
Collab History
Services
HCInvestigatorPhysicalButton
Static
Tag
Untagged
Layer
Default
Prefab
Select
Revert
Apply
Transform
Position
X -1.074344
Y 30.22986
Z 8.571336
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
HC Investigator Physical Button (Script)
Script
HCInvestigatorPhysicalButton
On Click ()
Runtime Only
No Function
HCInvestigatorPhysicall
Button Type
No Function
GameObject
Transform
HCInvestigatorPhysicalButton
bool enabled
string name
bool runInEditMode
string tag
bool useGUILayout
BroadcastMessage (string)
CancelInvoke (string)
CancelInvoke ()
RecordAudio ()
RecordTextData ()
RecordVideo ()
SendMessage (string)
SendMessageUpwards (string)
StopAllCoroutines ()
StopCoroutine (string)

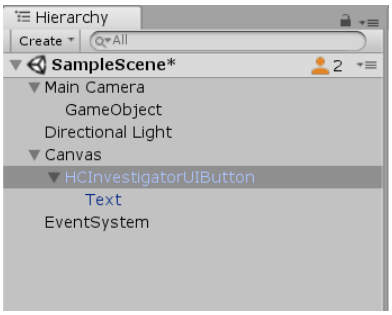
2 HCInvestigatorUIButton

This section goes over implementing the HCInvestigatorUIButton into a Unity project.

2.1 Creating Button Game Object

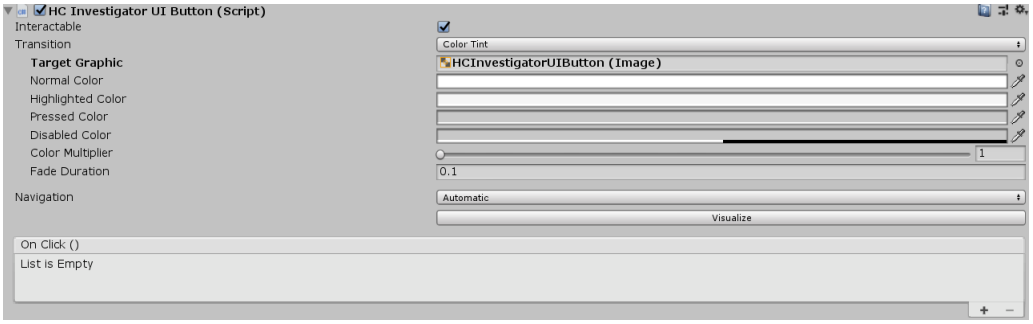
First, right click in the scene hierarchy, navigate to "UI", and click "Canvas". Then drag the HCInvestigatorUIButton prefab into the scene as a child of the Canvas.



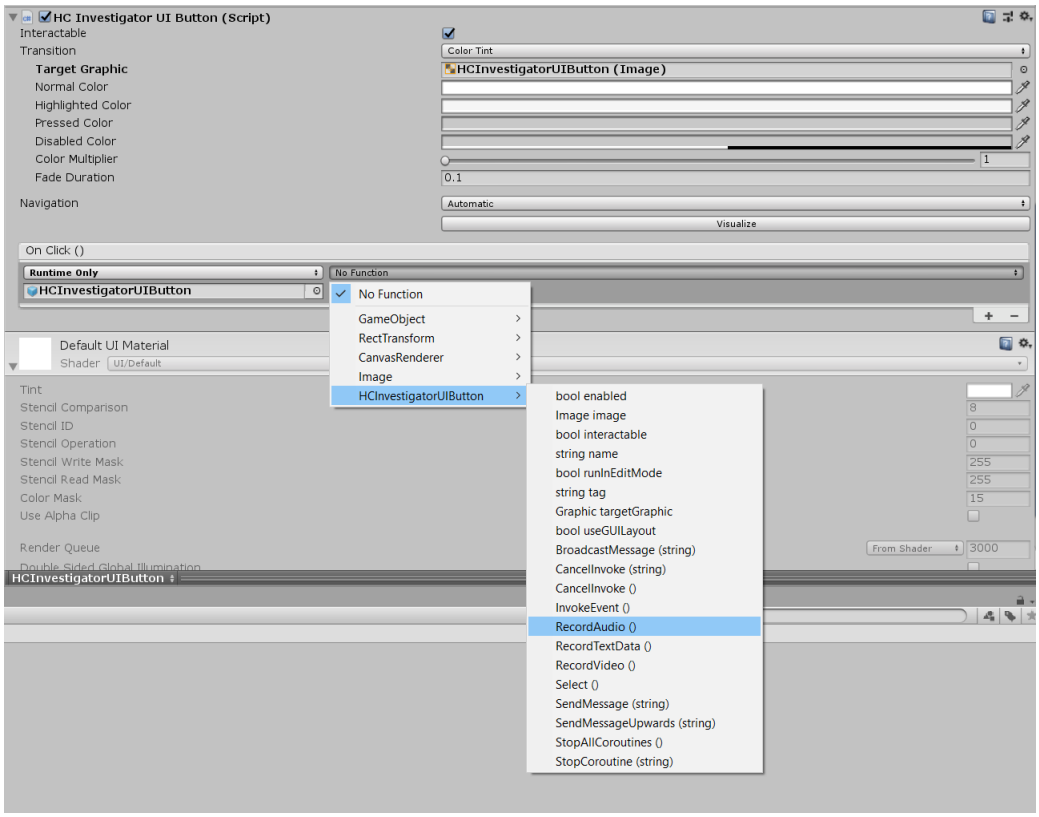


2.2 Adding Event to Button

Next, click on the button object in the scene hierarchy and navigating to the inspector. Add an event to the button's OnClick() function by pressing the "+", which is located under the HCInvestigatorUIButton (Script) section where it says OnClick().



Now, in the same section, click on the button game object in the scene hierarchy and drag it into where it says "None (Object)". You are now able to select which function you want to be called when the button is clicked. To do this, go back to the "OnClick()" section, click the drop down that says "No Function", navigate to "HCInvestigatorUIButton", and choose the desired functions (RecordTextData, RecordAudio, and/or RecordVideo).


HC Investigator UI Button (Script)
Interactable
Transition
Color Tint
Target Graphic
HCInvestigatorUIButton (Image)
Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier
Fade Duration
0.1
Navigation
Automatic
Visualize
On Click ()
Runtime Only
No Function
HCInvestigatorUIButton
No Function
GameObject
RectTransform
CanvasRenderer
Image
HCInvestigatorUIButton
bool enabled
Image image
bool interactable
string name
bool runInEditMode
string tag
Graphic targetGraphic
bool useGUILayout
BroadcastMessage (string)
CancelInvoke (string)
CancelInvoke ()
InvokeEvent ()
RecordAudio ()
RecordTextData ()
RecordVideo ()
Select ()
SendMessage (string)
SendMessageUpwards (string)
StopAllCoroutines ()
StopCoroutine (string)
Default UI Material
Shader
UI/Default
Tint
Stencil Comparison
Stencil ID
Stencil Operation
Stencil Write Mask
Stencil Read Mask
Color Mask
Use Alpha Clip
Render Queue
From Shader
3000
HCInvestigatorUIButton

3 Setting up HCInvestigatorManager

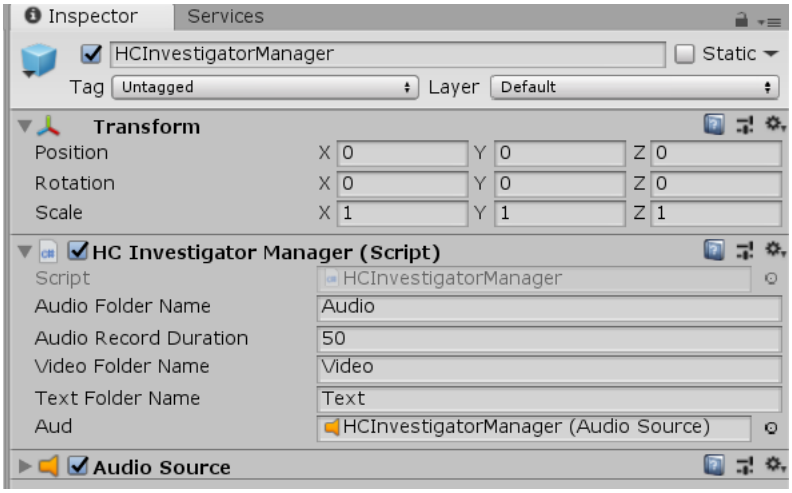
This section goes over implementing the HCInvestigatorManager into a Unity project.

3.1 Adding HCInvestigatorManager prefab

First add the HCInvestigatorManager prefab into the scene, this is done by navigating to the prefabs folder and dragging it into the scene.

3.2 Setting up the HCInvestigaotrManager variables

Edit the variables in the prefab, the audio folder name is the name of the folder that will hold audio recordings, the Audio record duration is how long the clip will be in seconds, the video folder name is the name of the folder holding the video recordings, the text folder name is the name for the text data logs and the Aud is the audio source.

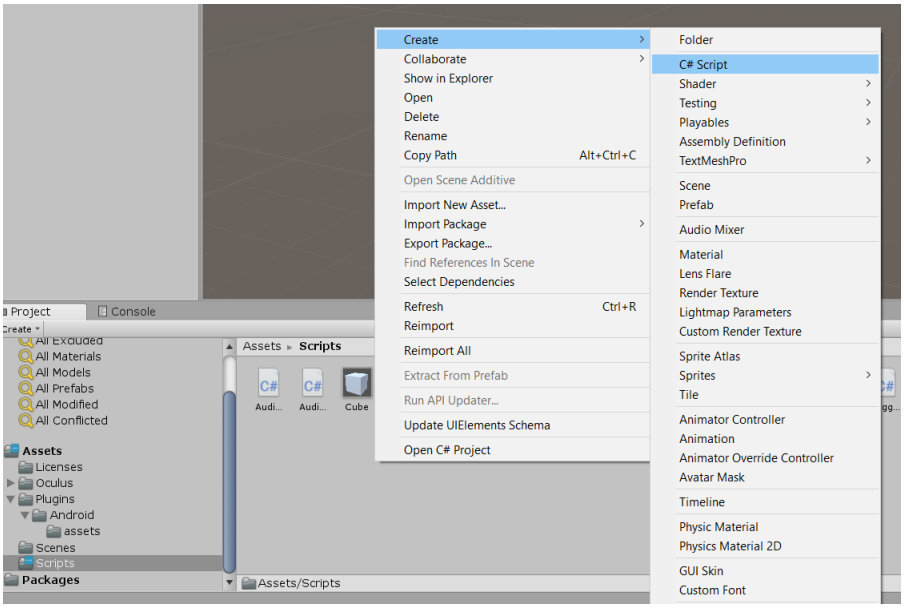


4 HCInvestigatorEvent

This section will go over setting up a HCInvestigatorEvent.

4.1 Make a new derived script

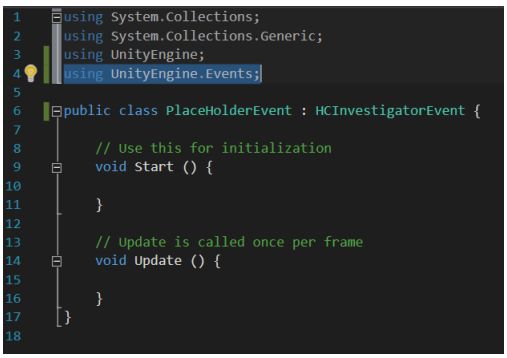
Create a new CSharp script by right clicking in the project view in whatever folder you want the script to be in and name it something like PlaceHolderEvent so it is easy to remember what the script does. Replace PlaceHolder with whatever the script is actually going to so. For example if I am monitoring heart rate I would call it HeartRateEvent.



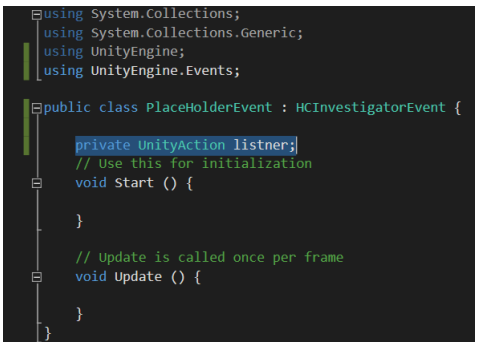
After creating the script open it up and replace MonoBehavior with HCInvestigatorEvent.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlaceHolderEvent : HCInvestigatorEvent {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

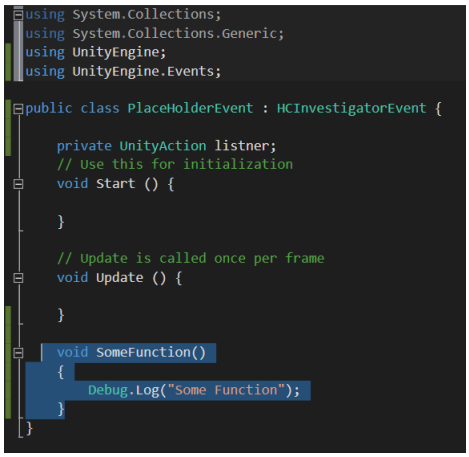
After that add using UnityEngine.Events to the top of the script



After that add a private UnityAction to the top of the class.



After that create function(s) for what you want this class to accomplish.



Next the function will need to linked to the Unityaction you made and will

need to be registered to the HCInvestigatorManager.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;

public class PlaceHolderEvent : HCInvestigatorEvent {

    private UnityAction listner;
    // Use this for initialization
    void Start () {

        listner = new UnityAction(SomeFunction);
        RegisterToManager(listner);
    }

    // Update is called once per frame
    void Update () {

    }

    void SomeFunction()
    {
        Debug.Log("Some Function");
    }
}
```

Once this is done you have almost all the necessary code set up to add an event to the scene. If your code needs to have more than one function in it to be called just add the same number of UnityActions and link them up. So for every function it gets one unity action.

The next code related thing your event needs is a trigger to call the event this can be a number of things from a certain button input to an object colliding with something. To set this up open the HCInvestigatorManager and add a function that will Invoke the event.

```
    public void InvokeEvent(string eventName)
    {
        if (!eventDict.ContainsKey(eventName))
        {
            Debug.LogError("Key doesnt exist in dictionary");
        }
        else
        {
            eventDict[eventName].Invoke();
        }
    }
}
```

4.2 Add a function that invokes the event

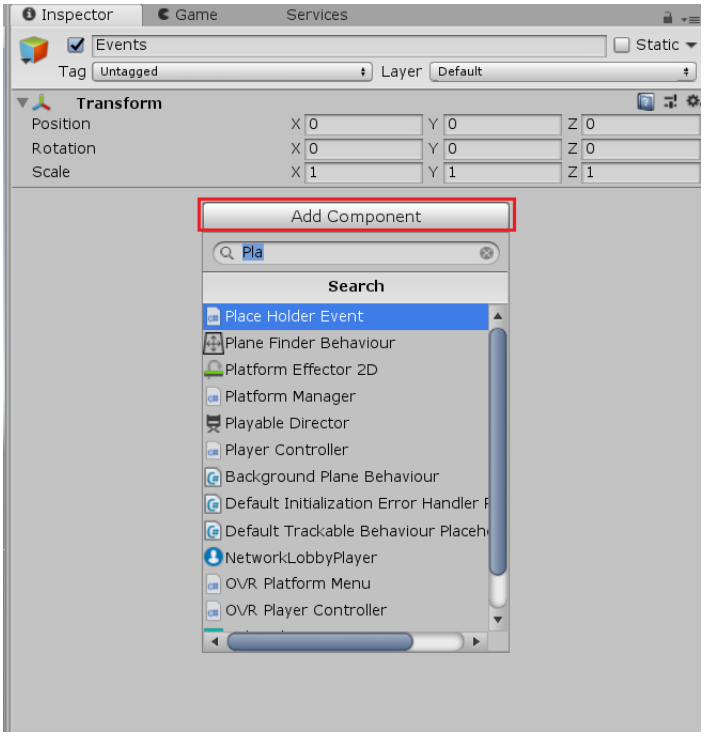
The last thing you will need to do codewise is call the InvokeEvent Function in the place you want the event to trigger. I will show you how to do it from pressing the P key on the keyboard in the HCInvestigatorManager Script.

```
void Update()
{
    if(Input.GetKeyDown("p"))
    {
        InvokeEvent("test");
    }
}
```

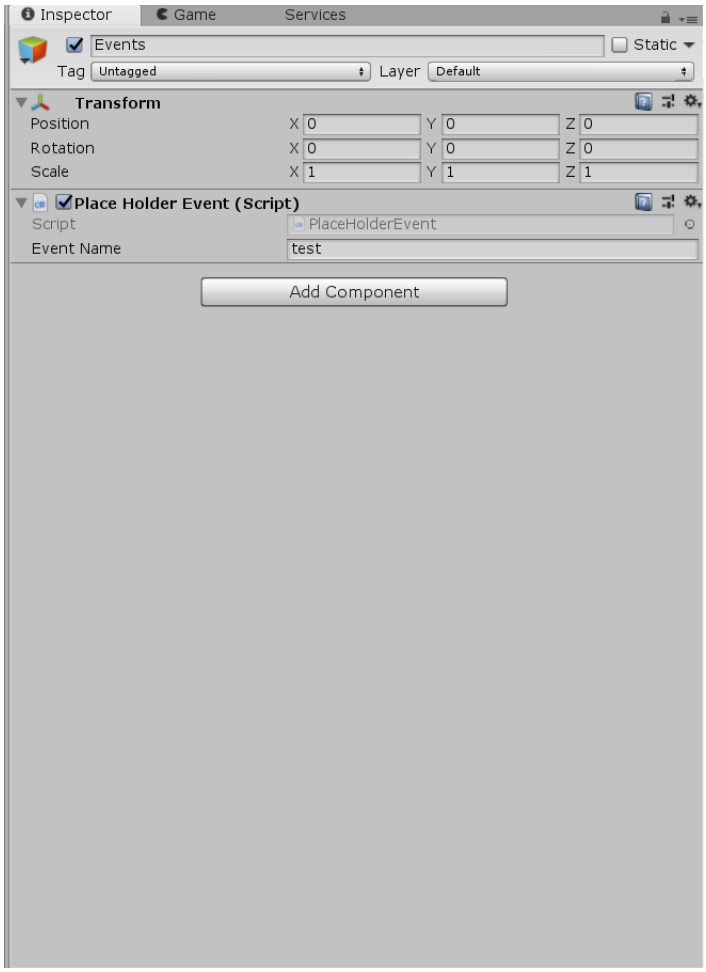
This will call the test event that I will set up in the inspector in the next step.

4.3 Adding a HCInvestigatorEvent to the scene

The next step to set up a HCInvestigatorEvent is adding it to the scene. Pick which game object you want the script on and add it to it in the inspector.



After it is added the last thing you need to do is choose what name you want for the event.


Inspector
Game
Services
Events
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
X 0
Y 0
Z 0
X 0
Y 0
Z 0
X 1
Y 1
Z 1
Place Holder Event (Script)
Script
PlaceHolderEvent
Event Name
test
Add Component