Leap Motion and Unity

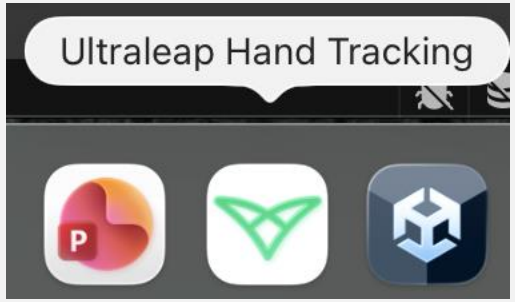
Corso Realtà Virtuale 2025/2026

eleonora.chitti@unimi.it

https://github.com/aislabunimi/courses.vr2026

Ultraleap Hand Tracking

1. To check Leap Motion service you can open on your PC the application:



Exercise 1

A first Leap Motion project together

First action to do

1. Create a new 3D scene in Unity (choose Universal pipeline)

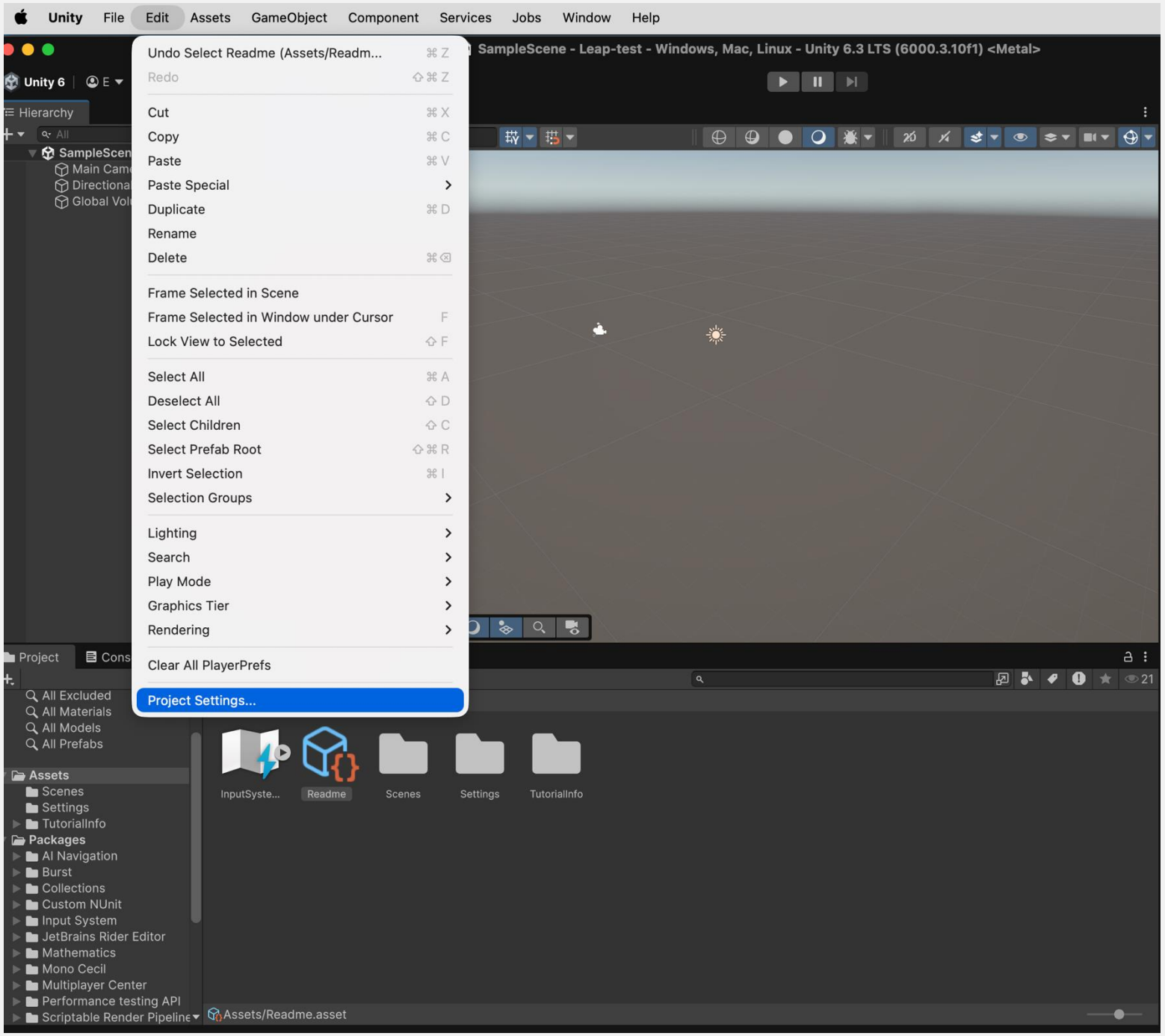
n.b. LeapMotion is compatible with both Unuversal -URP and Standard pipelines

Setup the Unity Project

https://docs.ultraleap.com/xr-and-tabletop/xr/unity/getting-started/index.html

First Steps

1. Go to Project Settings



First Steps

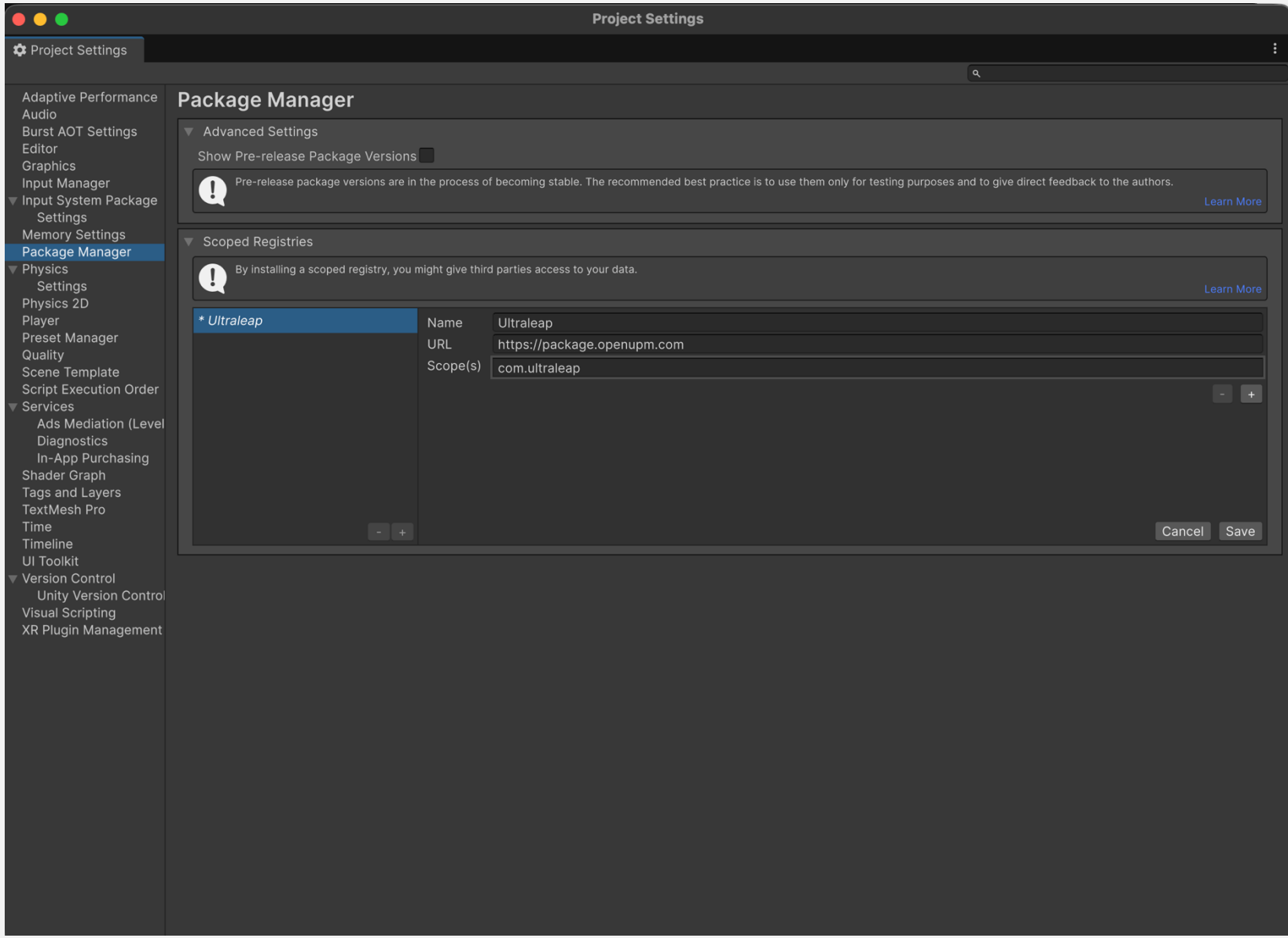
1. Click on Package Manager

2. Add Ultraleap and save:

Name: *Ultraleap*

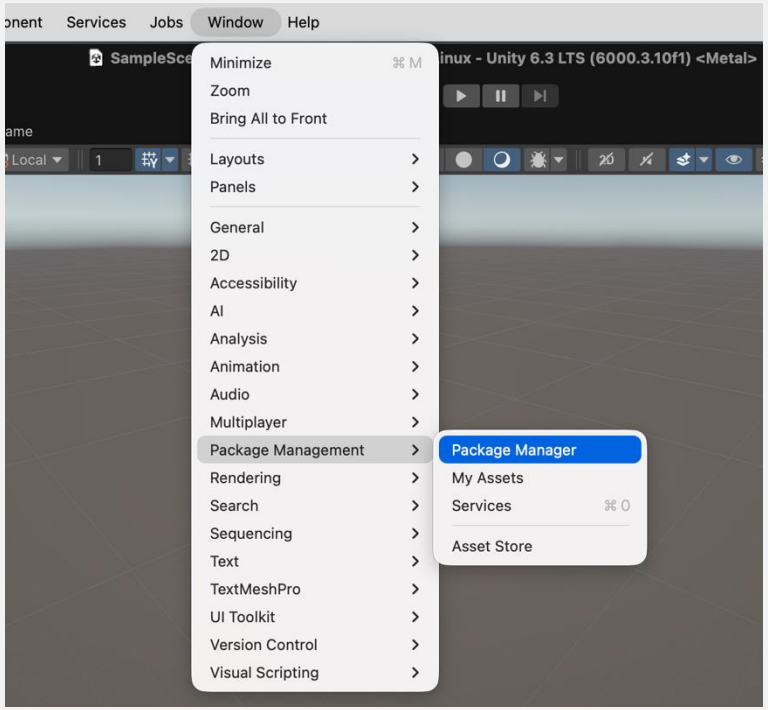
URL: *https://package.openupm.com*

Scope(s): *com.ultraleap*

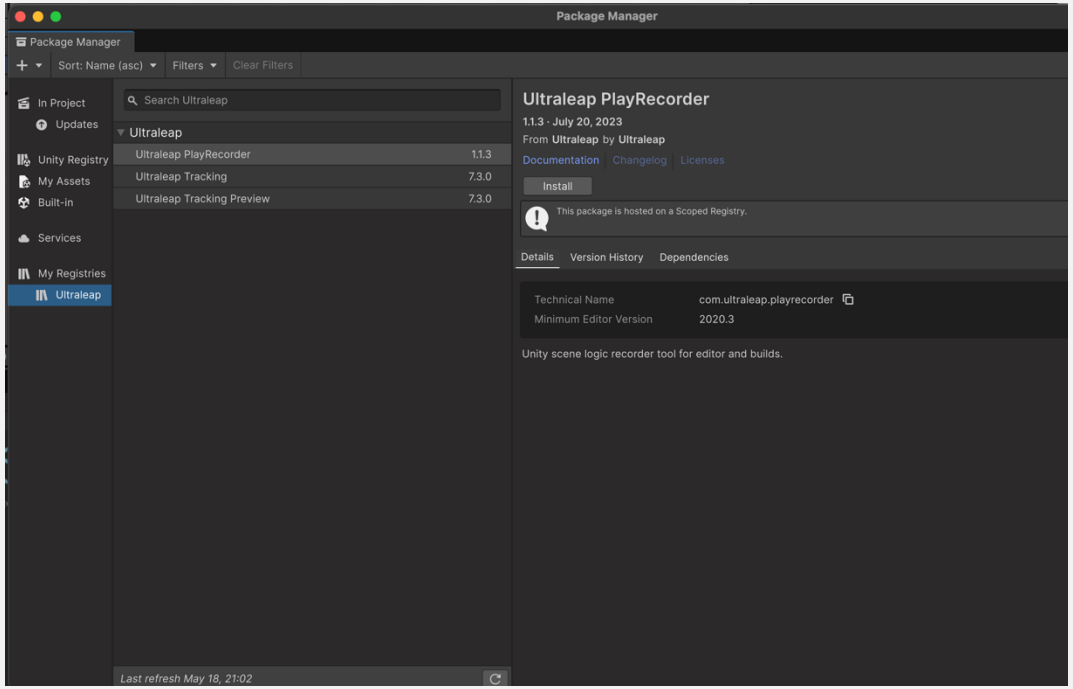


First Steps

1. Open the Package Manager (Window -> Package Manager)

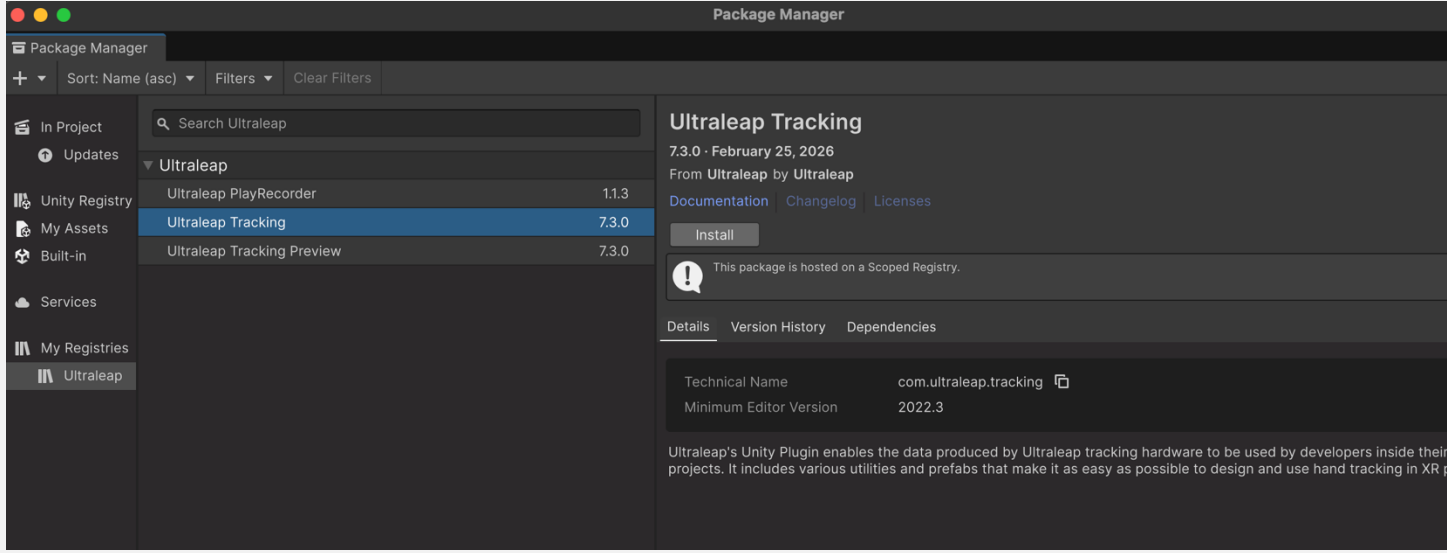


2. Navigate to “My Registries” in the menu at the left of the window, and click on Ultraleap.

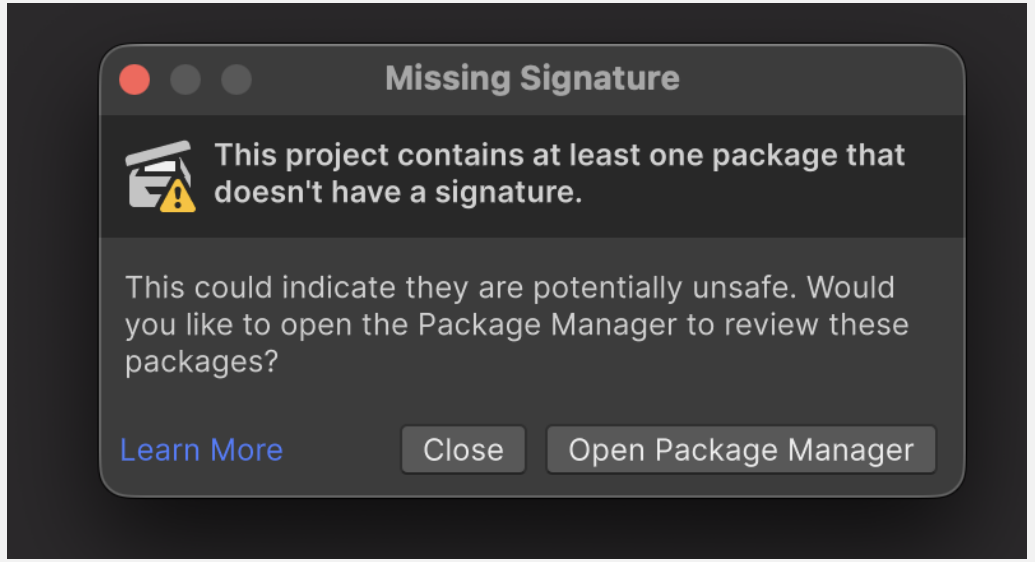


First Steps

1. Install only Ultraleap Tracking package (for now)

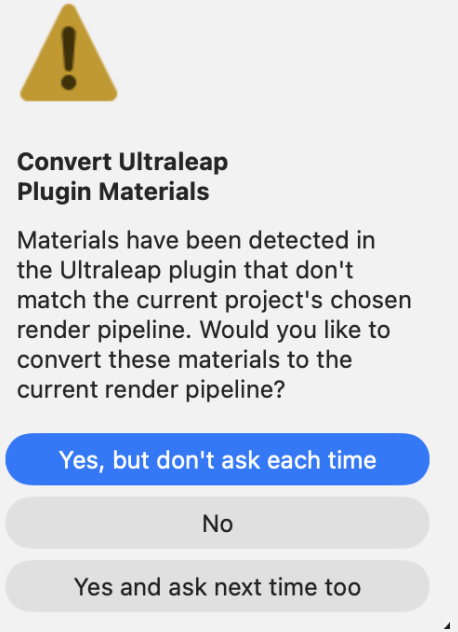


2. A popup may appear, click on open Package Manager



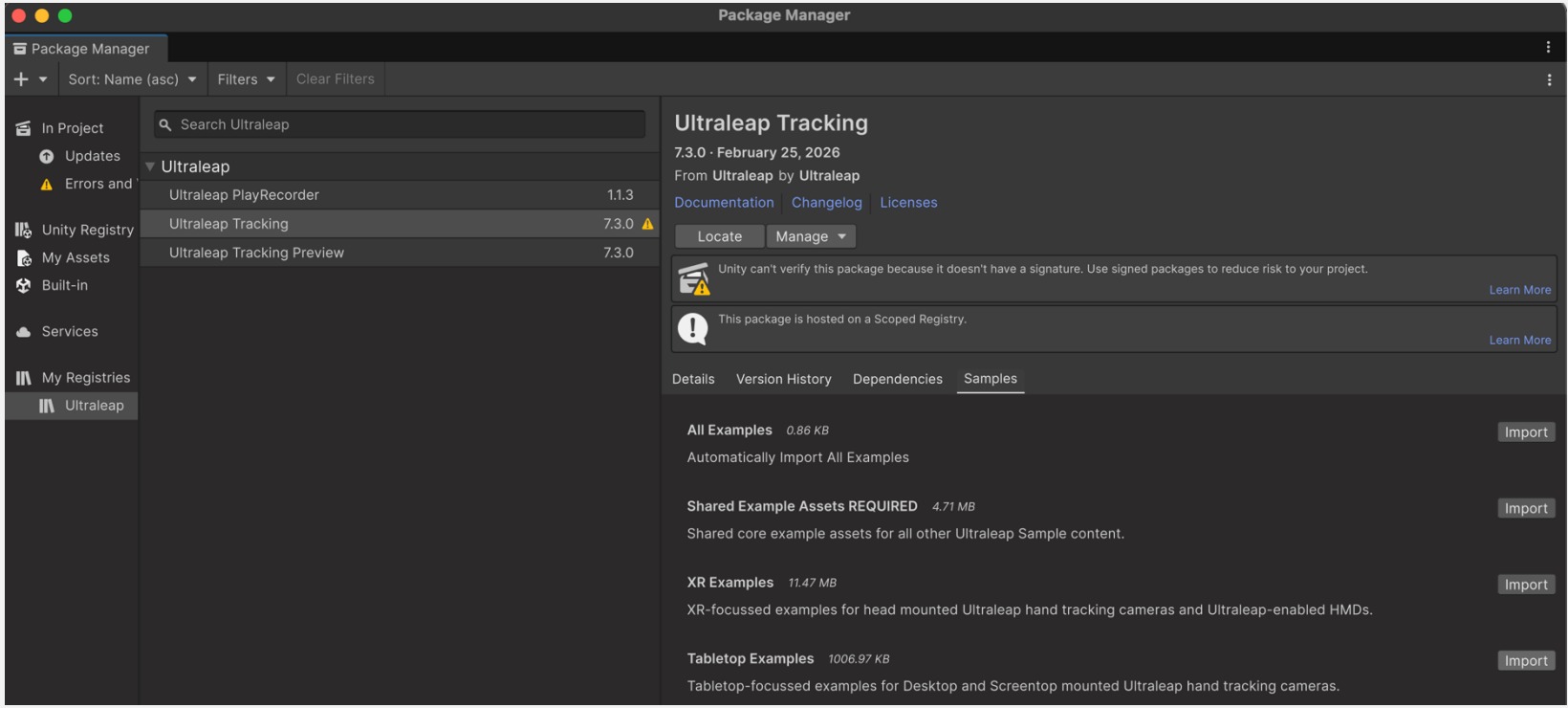
First Steps

1. Another popup may appear, say yes



2. Now go again in the Package Manager on Ultraleap and click on Samples

3. Now click on Install Tabletop Samples

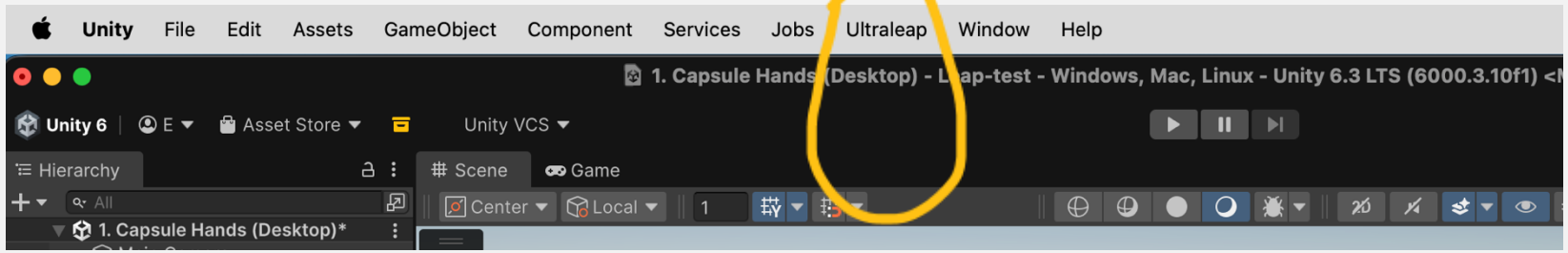


Work on the Scene with Leap

https://docs.ultraleap.com/xr-and-tabletop/xr/unity/getting-started/your-first-project.html

Setup Scene

- Now a Ultaleap button appeared in the toolbar



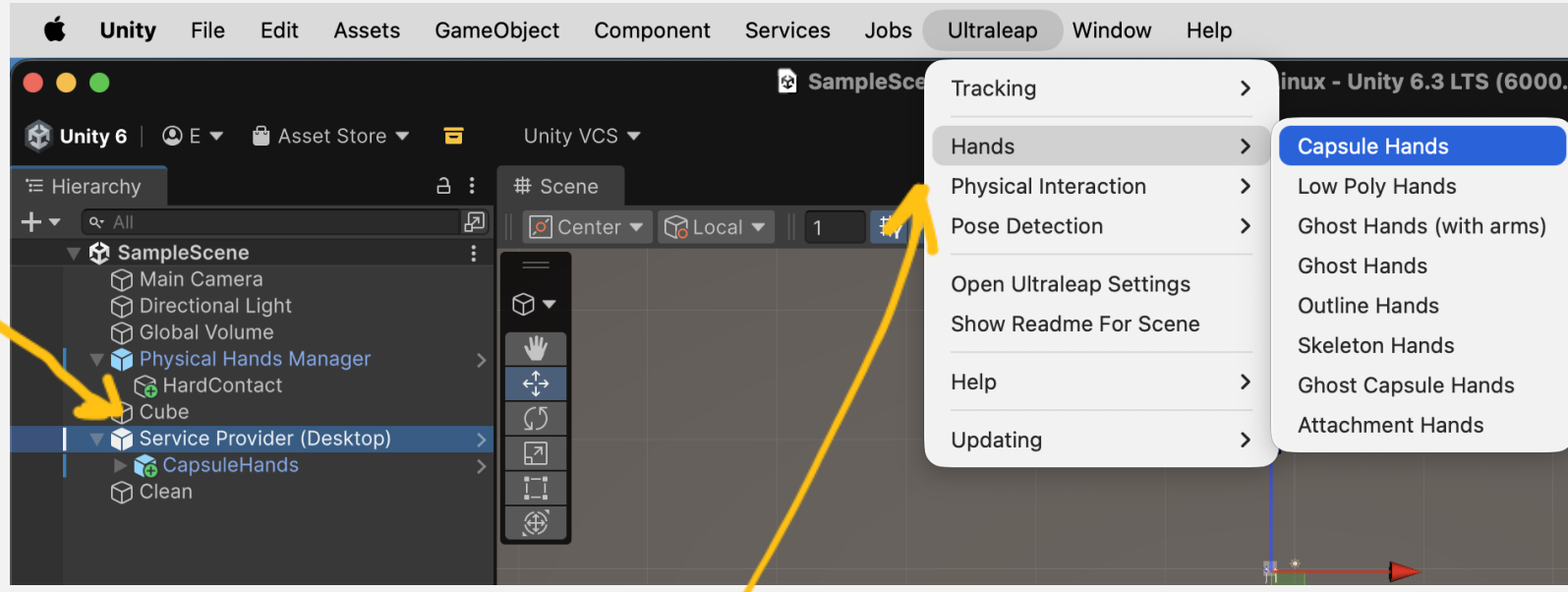
- In the toolbar click on *Ultraleap* > *Tracking* > *Sercive Provider (Desktop)* This action adds an XR Leap Provider Manager prefab to the scene. This provides hand tracking service data to the Unity application.

Setup Scene

Click on the *Service Provider (Desktop)* in the Scene

From the same toolbar menu as before, add a set of hands to the scene *Ultraleap* > *Hands* (it will be added as a child of the XR Leap Provider Manager since it is selected). The plugin provides different hand visuals, add one you like.

Now Press Play and the tracking will work

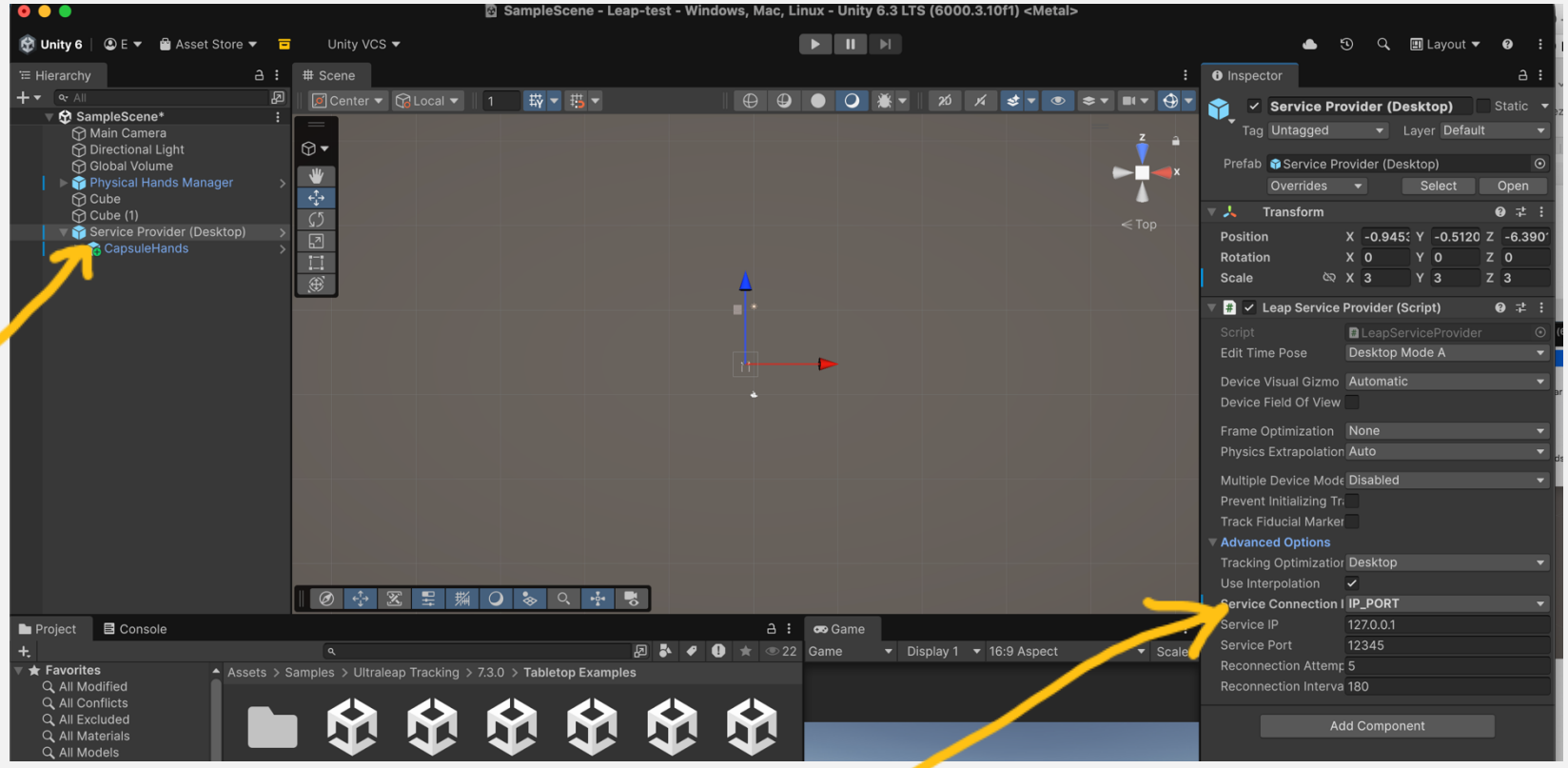


Troubleshooting

If Leap Service is not found

1) Check if service is running using the Ultraleap Hand Tracking application to test

2) In the Scene in Unity: click on the *Service Provider* and in the *Inspector* click on the *Advanced Option* and change Service Connection with IP_PORT



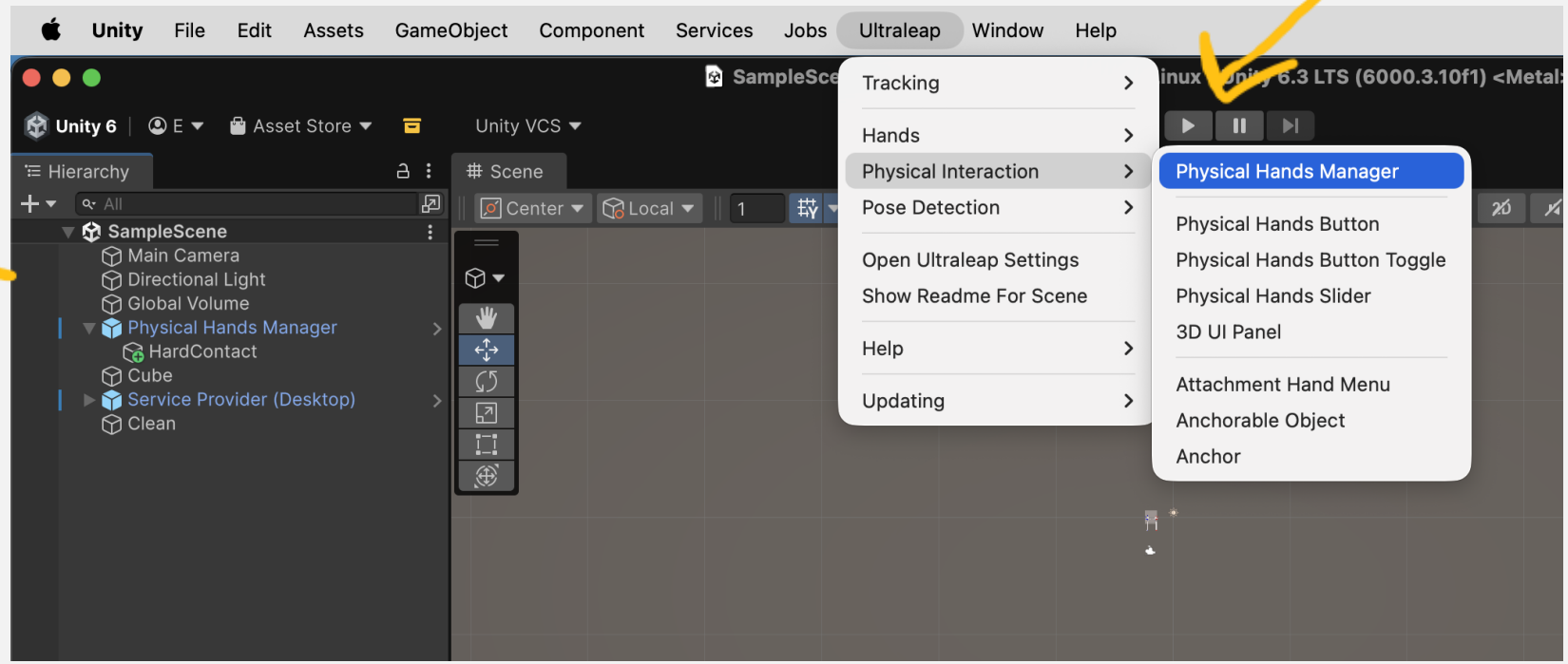
Hand Physical Interactions

https://docs.ultraleap.com/xr-and-tabletop/xr/unity/plugin/features/physical-hands.html

Setup Scene with Leap Prefab for interaction

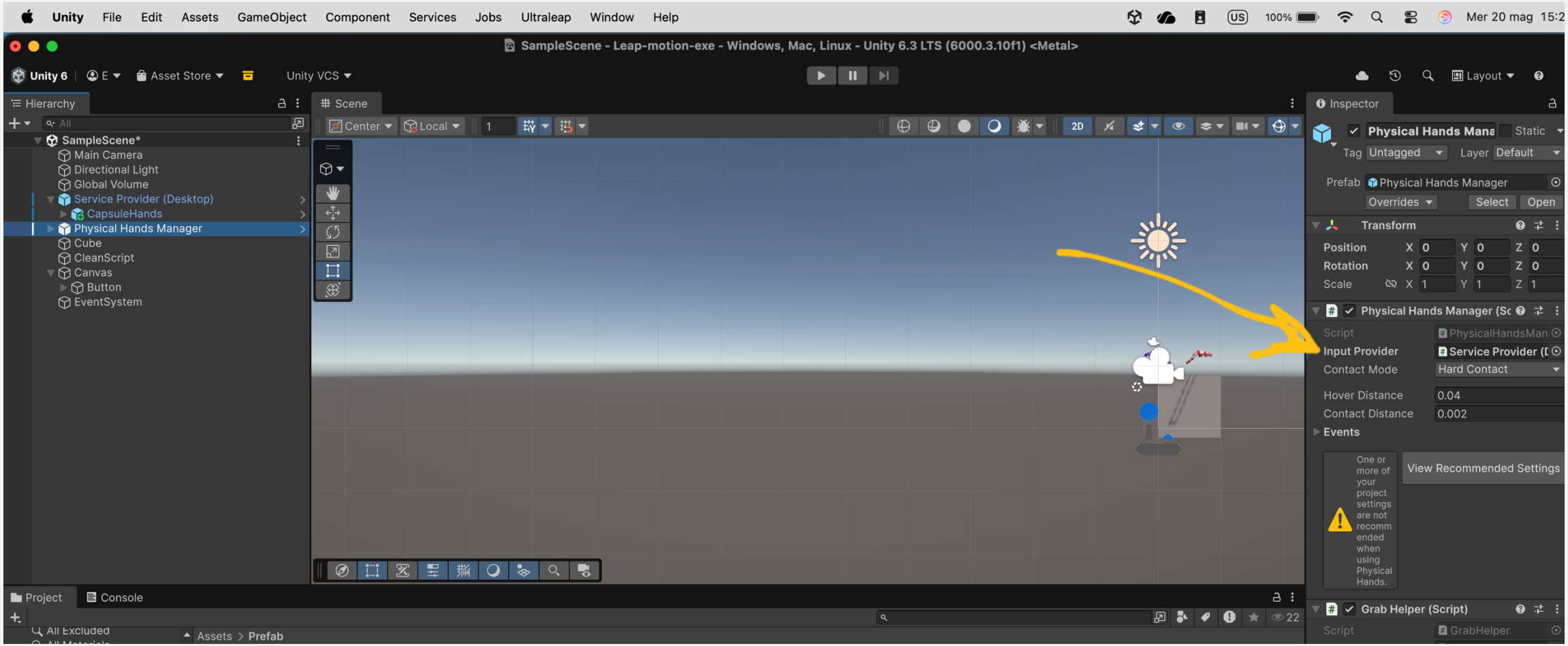
- Deselect the *Service Provider (Desktop)* in the Scene (<u>we do not want</u> to add this component as child)

- Go and click on *Ultraleap* > *Physical Interaction* > *Physical Hands Manager* to add it in the scene



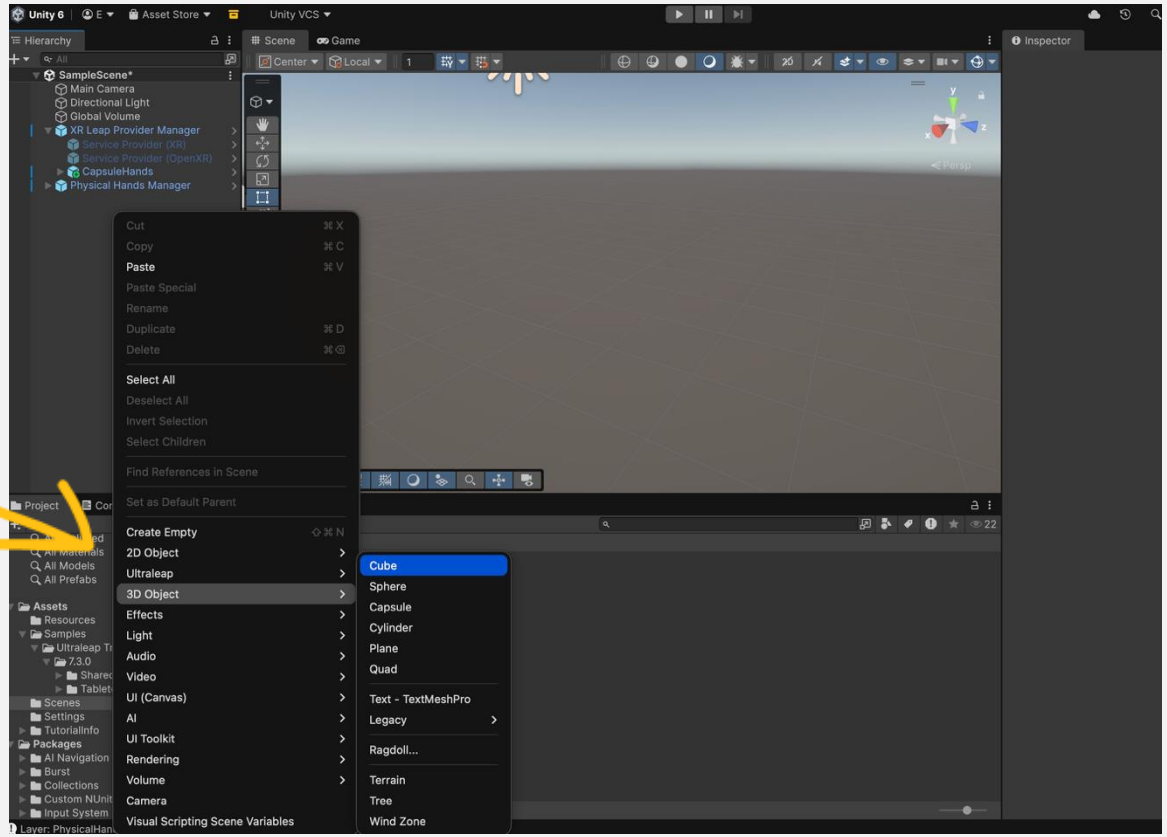
Setup Scene with Leap Prefab for interaction

- In the inspector of *Physical Hands Manager* add the *Service Provider (Desktop)* as input provider



Add a cube

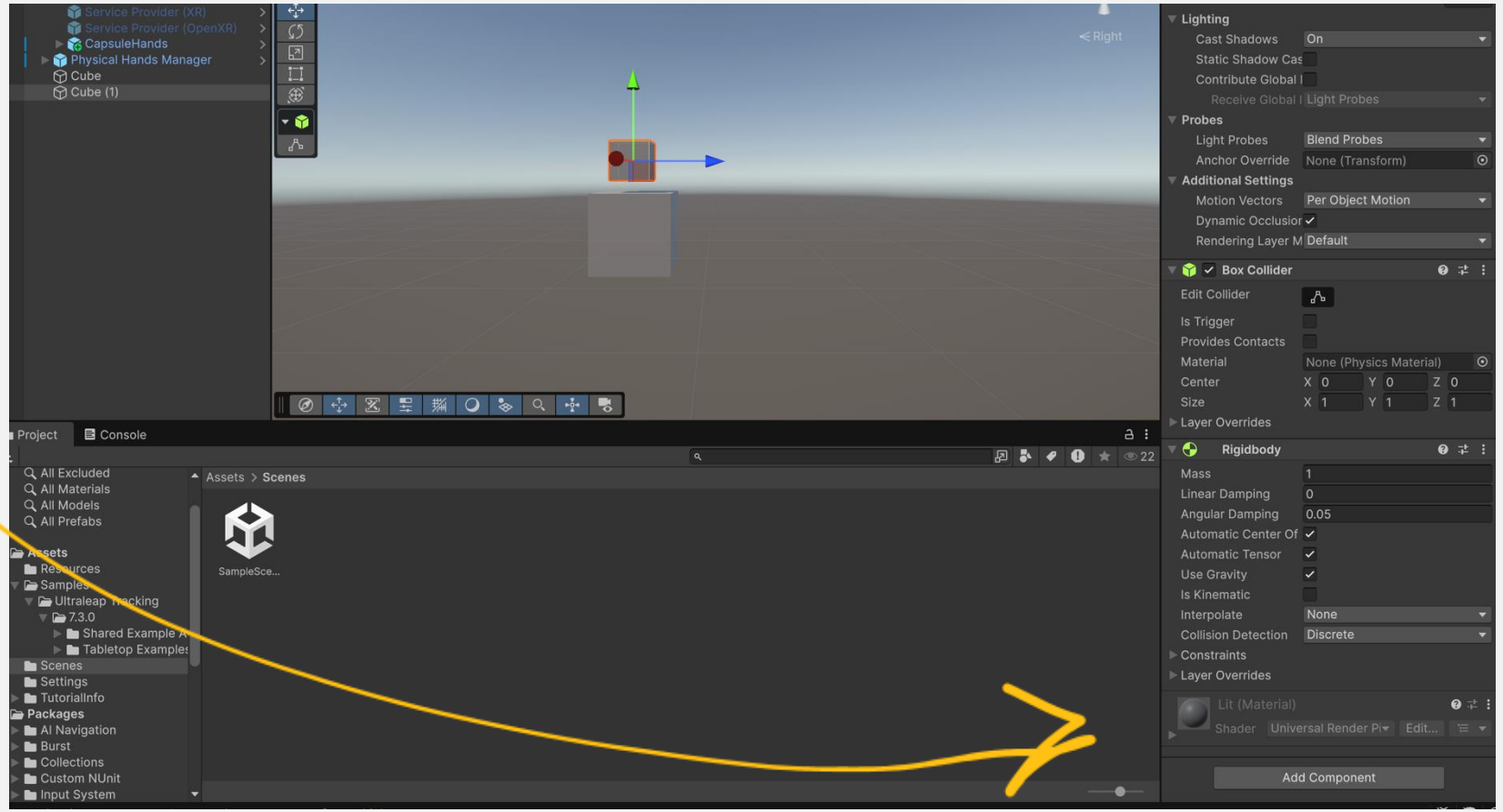
- Add a Cube in the Scene (the cube will have automatically a box collider)
- Add another cube and scale it (reduce its size), put this cube on top of the first cube



Add rigidbody

Add a rigidbody to the smaller cube

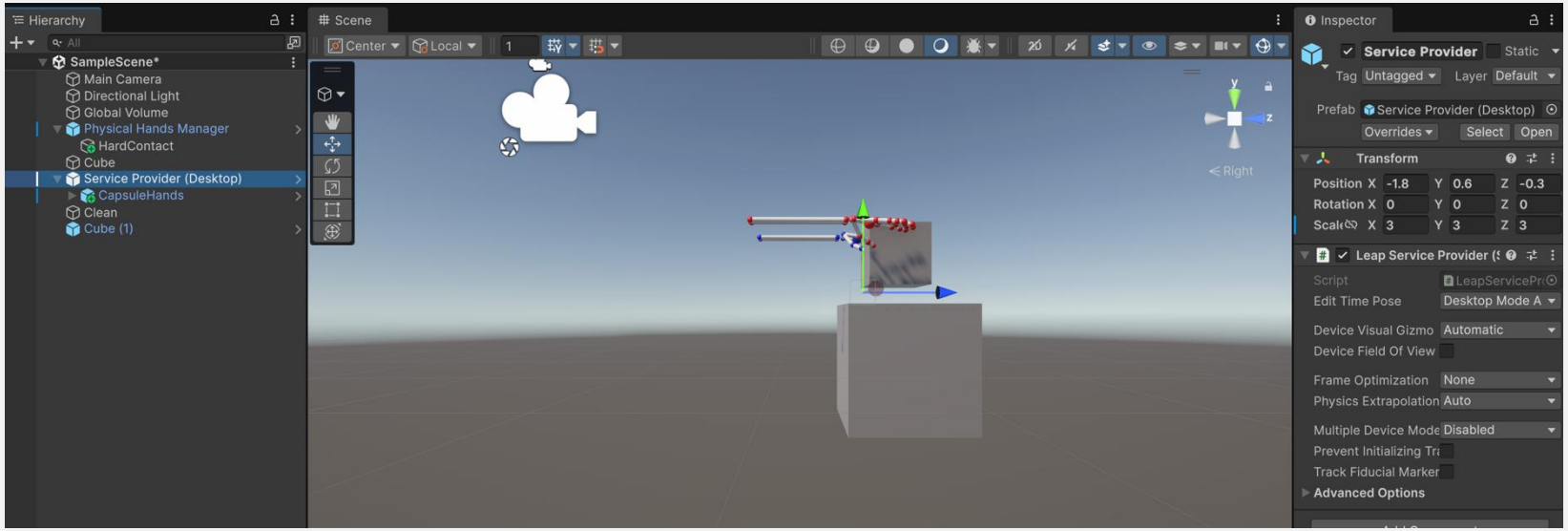
Now you can play and test your application



Adjust the scene and play!

You can adjust the position of Service Provider Desktop so that the hands appear near the cube

You can also adjust the size of the Service Provider Desktop in order to change hands' size

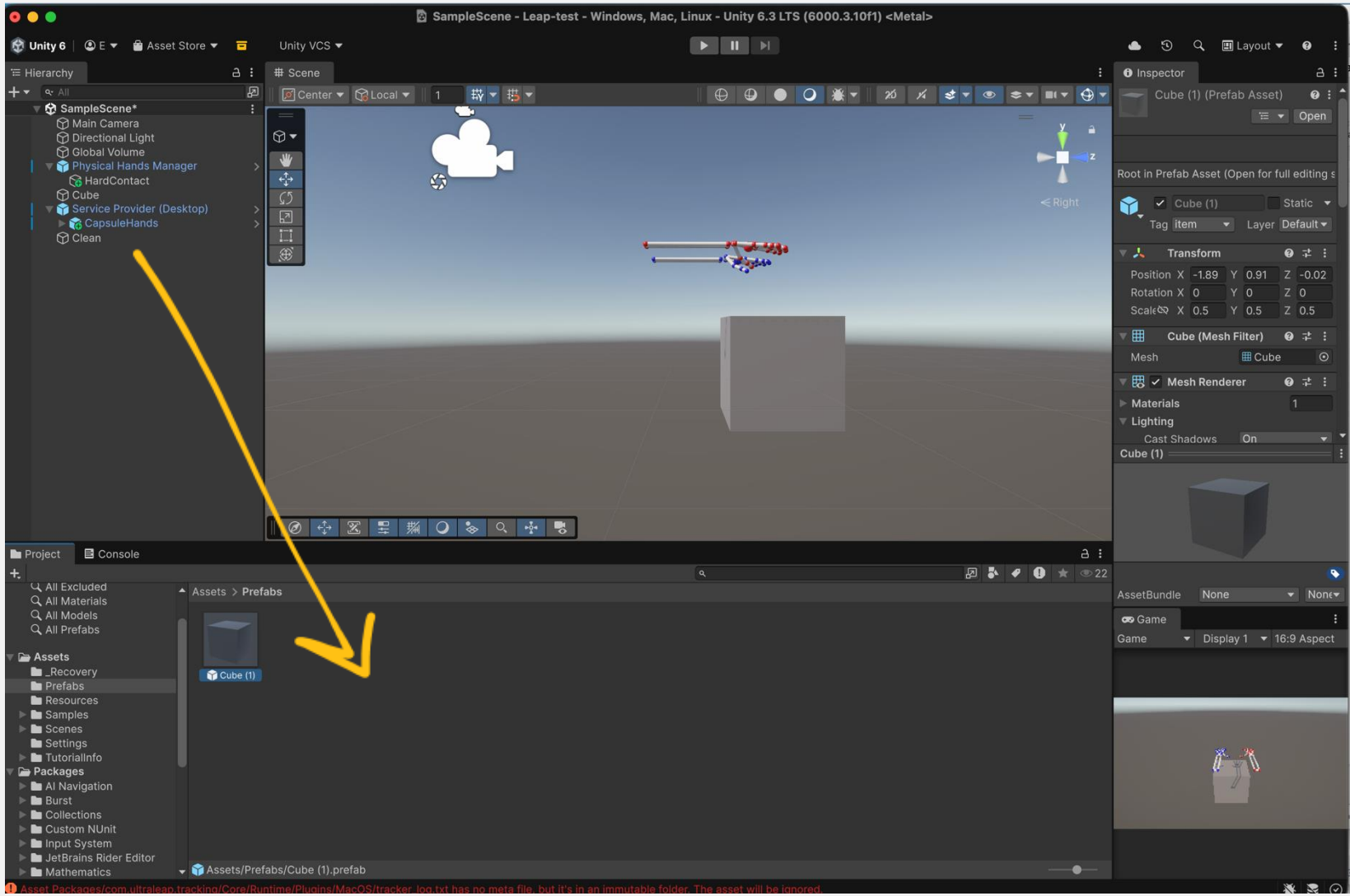


Ooops the cube fell down

1. Create a Prefab of the Cube with Rigidbody

2. Delete the Cube with Rigidbody from the scene

3. Create a script CleanScene and write in the script the code in the next slide



CleanScene

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

public class CleanScene : MonoBehaviour
{

    public List<GameObject> items;

    // Start is called once before the first execution of Update after the MonoBehaviour is created
    void Start()
    {
     CleanAndSetup();
    }

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

    }

    public void CleanAndSetup()
    {
        GameObject[] objects = GameObject.FindGameObjectsWithTag("item");
        foreach (GameObject i in objects){
            Destroy(i);
        }
        foreach (GameObject item in items)
        {
            Instantiate(item);

        }

    }
}
```

Takes a **public** list of GameObjects (you can add them from the Inspector in Unity) here we will insert the Prefab of the cube

Deletes all the fallen items

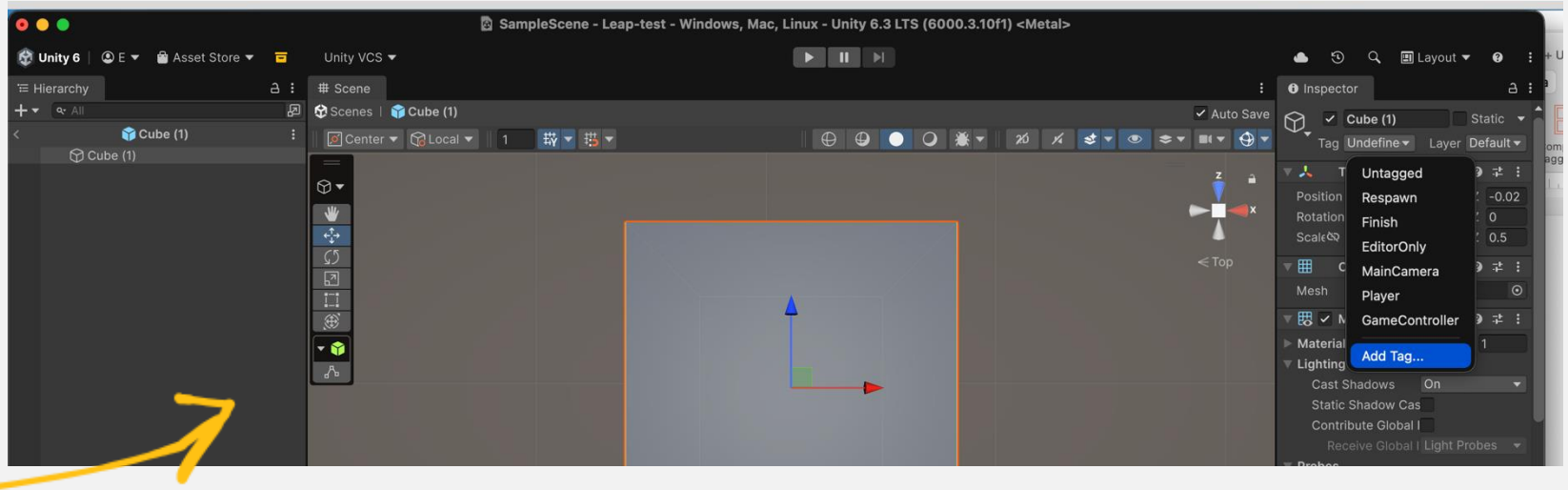
And Instantiates in the original position all the new items

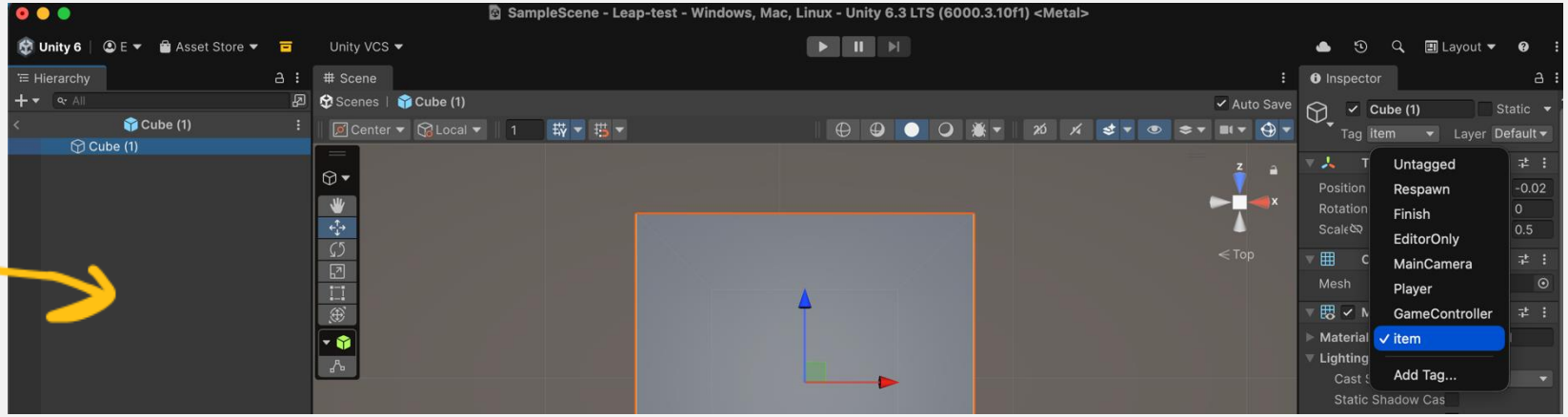
Ooops the cube fell down

1. Open the Prefab (Double click on the prefab)

2. Click on Add Tag, create a tag “item”

3. Click again on the prefab of the cube and add the tag Item



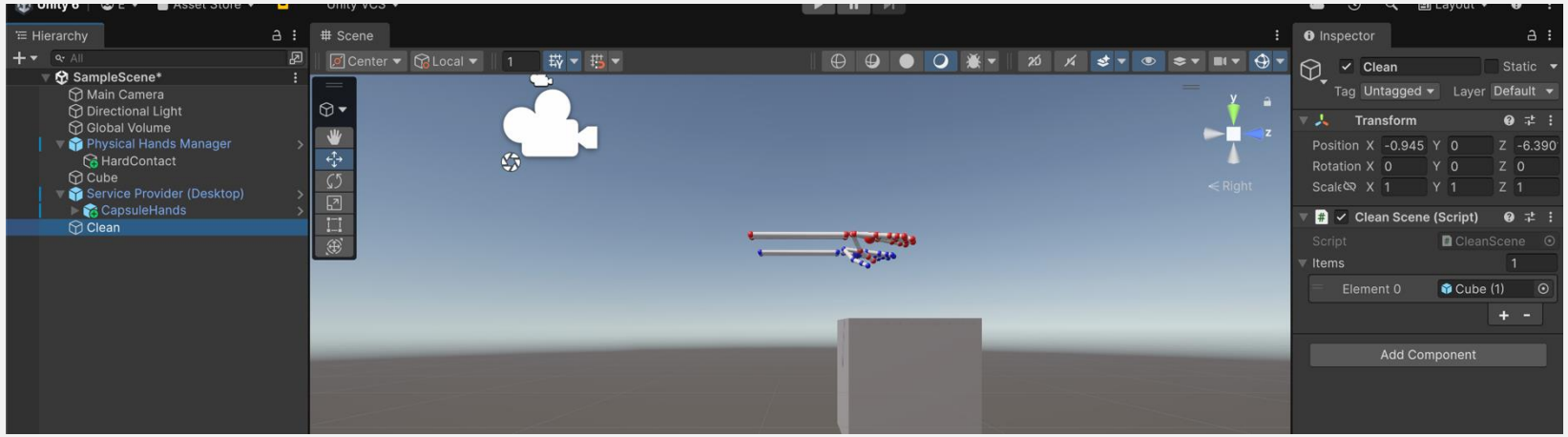


Ooops the cube fell down

1. In the scene create an Empty Game Object and call it “Clean”

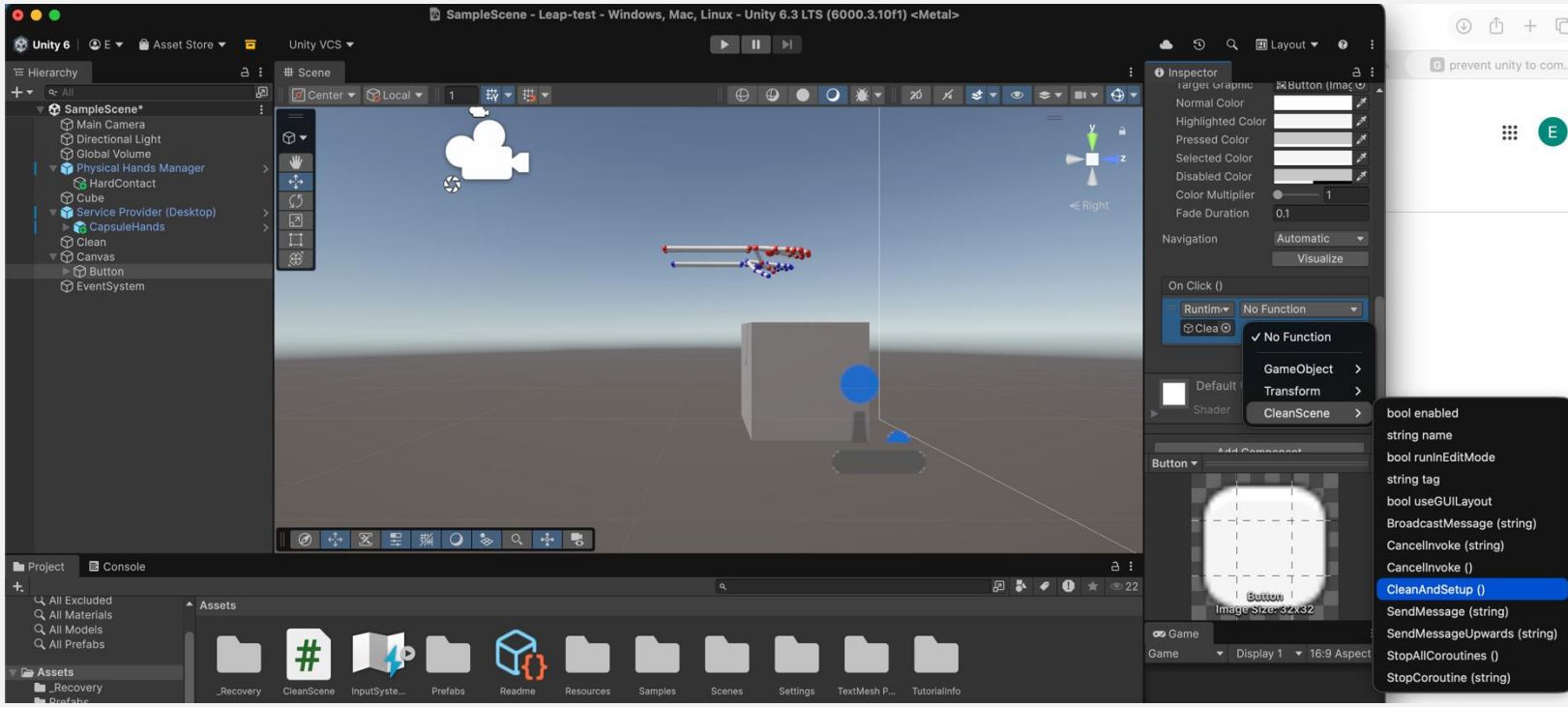
2. Add the CleanScene script to the Clean GameObject

3. Add the Prefab of the Cube in the List



Ooops the cube fell down

1. Add a Canvas to the scene
2. Add a button in the Canvas
3. Add the function CleanAndSetup to the button



Now you can reset the cube position

Leap Motion Unity exercise and homework

Corso Realtà Virtuale 2025/2026

eleonora.chitti@unimi.it

https://github.com/aislabunimi/courses.vr2026

Add fruit

In the Unity Project of the Cube we just made, import the **fruitpackage.unitypackage** that you can find in https://github.com/aislabunimi/courses.vr2026 in the folder Material_LeapMotion

Add at least 3 fruits in the scene

Put the fruit in the Baskets

Open the scene Food Scene and set it up as a Leap Motion scene (as we did in the first exercise):

1. Add the Leap Service Desktop
2. Add the Hands under the Leap Service
3. Add the Physical Hands Manager
4. Add a rigidbody to each fruit
5. Adjust the position and Size of the Leap Service
6. Make each fruit a prefab and tag it as “item”
7. Add the CleanScene script in a new empty Game Object
8. Add the Canvas and Button
9. Just Play!

Caricare su upload un file di testo con:

Nome Cognome dei partecipanti al gruppo

Link per scaricare il Progetto unity