

# Plugin integration Example

## Using UNET

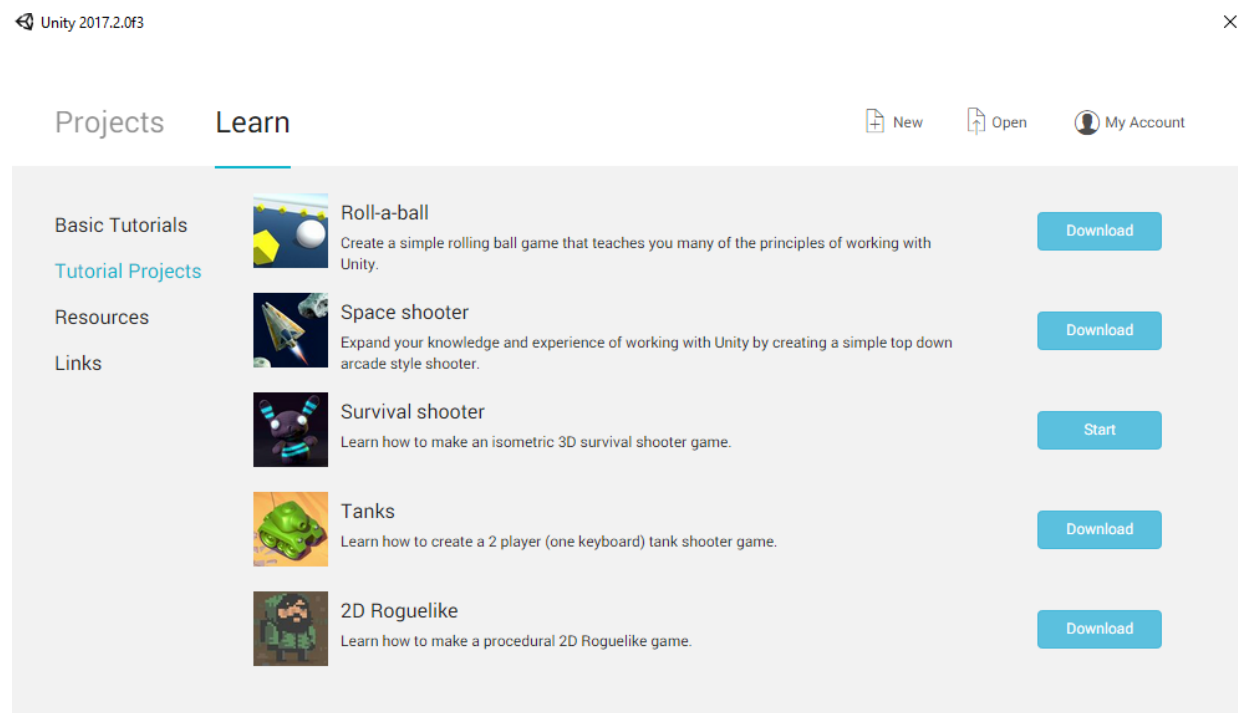
We are going to show how to integrate VR Tracker plugin in an existing Unity Project. You will see how to do it on the example project of Unity called Survival Shooter.

### 1. Download the plugin asset

You will find the latest version of the plugin here : <https://github.com/VR-Tracker/Plugins>

(We are going to use the file in the folder package, that can be easily imported to any project)

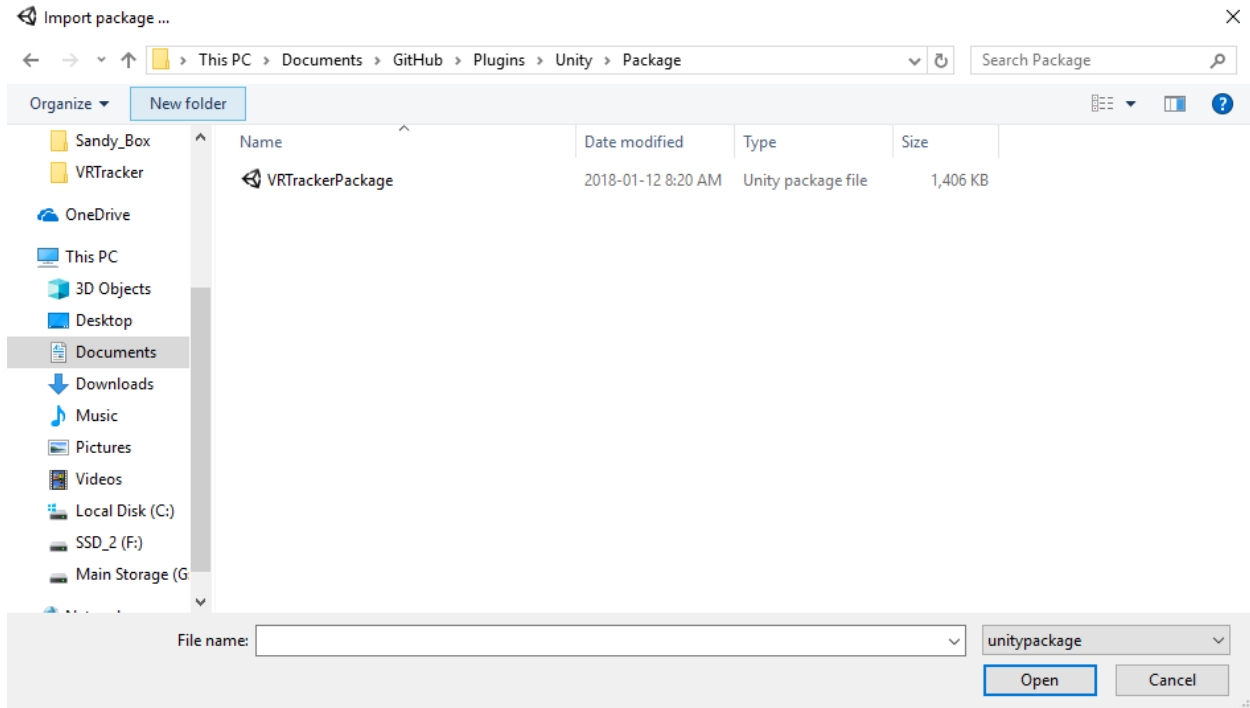
### 2. Open your Unity project



*1. Select your project*

Here we choose to use an existing project from the Unity Tutorial Projects

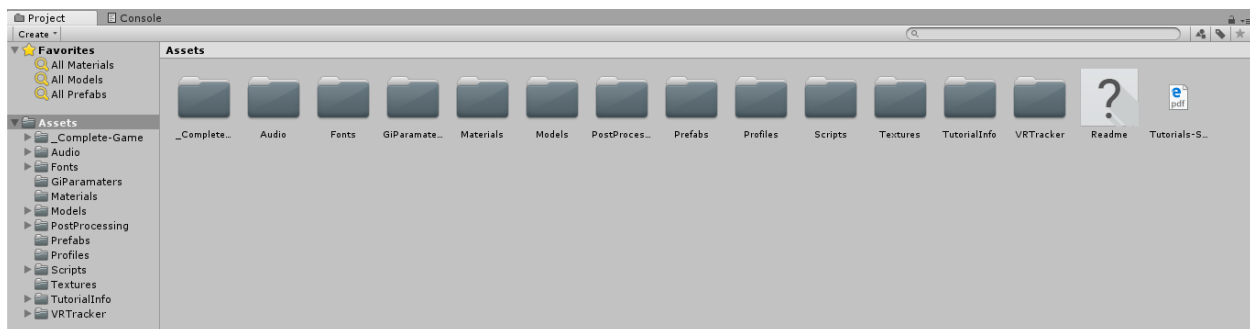
### 3. Import the VR Tracker plugin asset



*2.Assets>Import Package->Custom Package*

Choose the **VRTrackerPackage** that contains the VR Tracker Plugin

Click on import, you will need all the selected elements, after the importation you will be able to see a new folder VR Tracker in the Assets



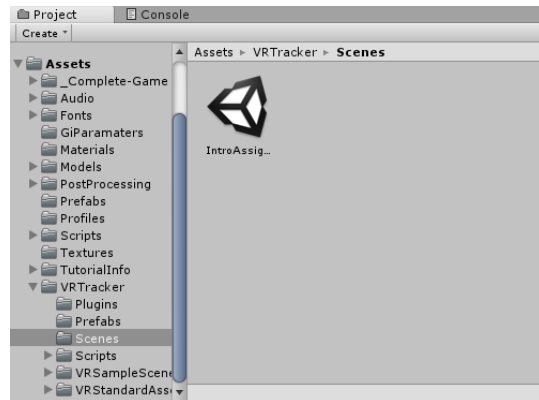
*3. VRTracker plugin components*

The VR Tracker folder contains the different sub-folders:

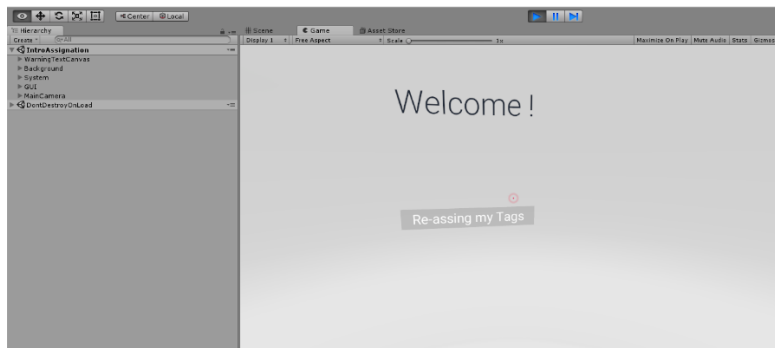
-Plugins, Prefabs, Scenes, Scripts, VRSampleScenes, VRStandardAsset

*(Do you think I should add the description of each folder here ?)*

In the Scenes sub-folder, we have one scene called IntroAssination that contain the tag association part and is one that you need to put before any other scene.



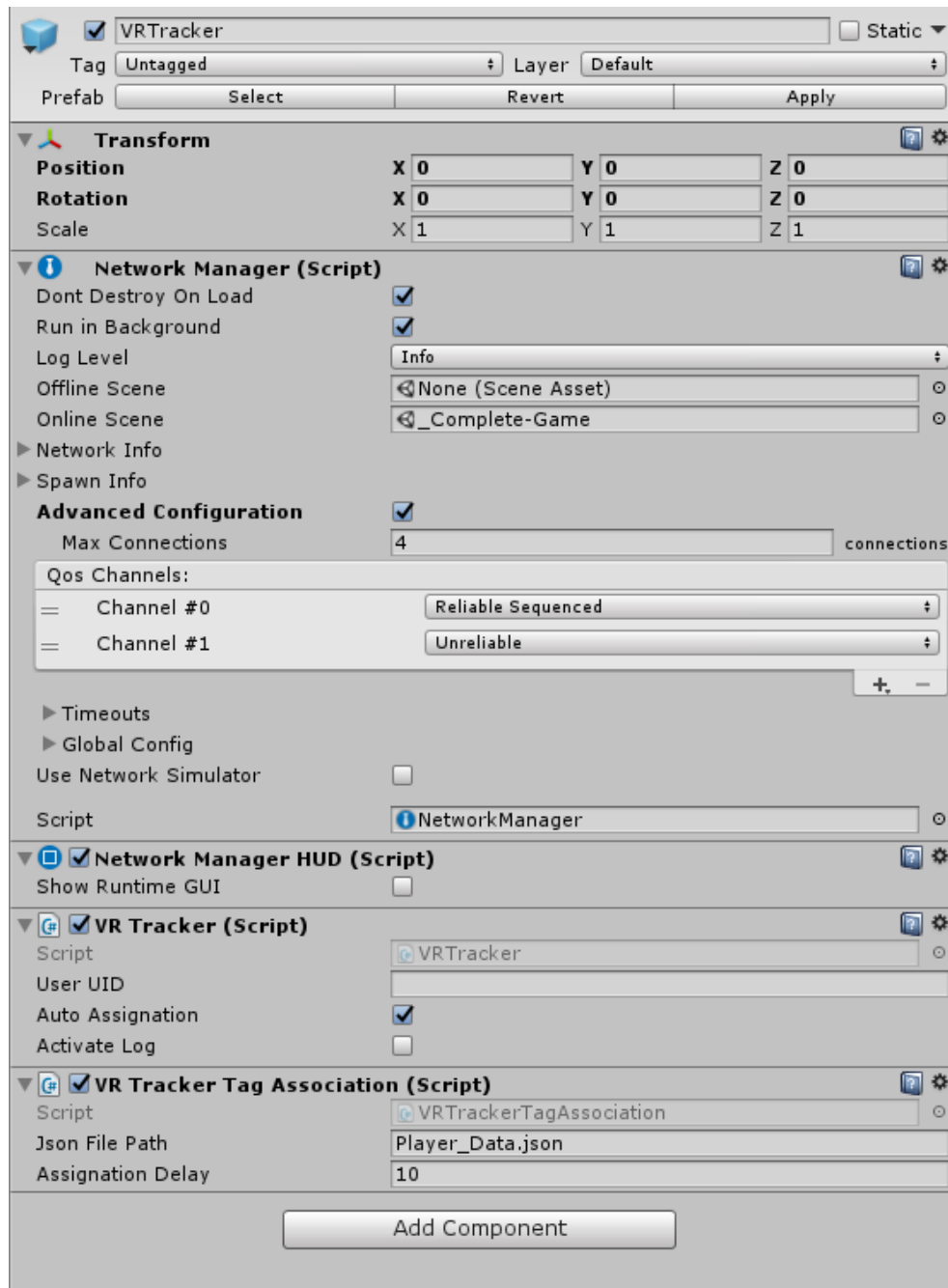
If you try to play the scene you should see directly something like this :



You won't have anything if you assign your tag as the is not element on it.

#### 4. Adding your own scene

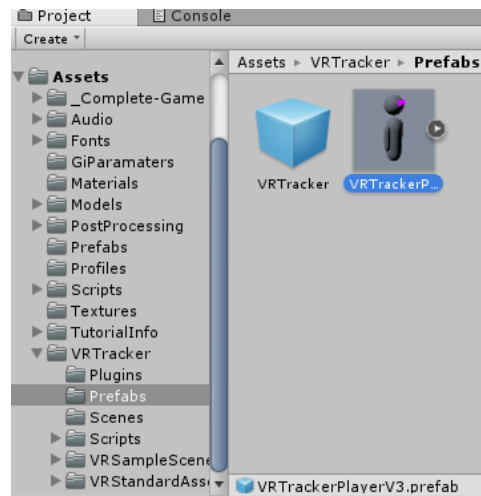
On the IntroAssignment Scene, you will find a *GameObject* called **VRTracker**, which contains the UNET Network Manager. You will be able to see in the inspector the following information when clicking on it.



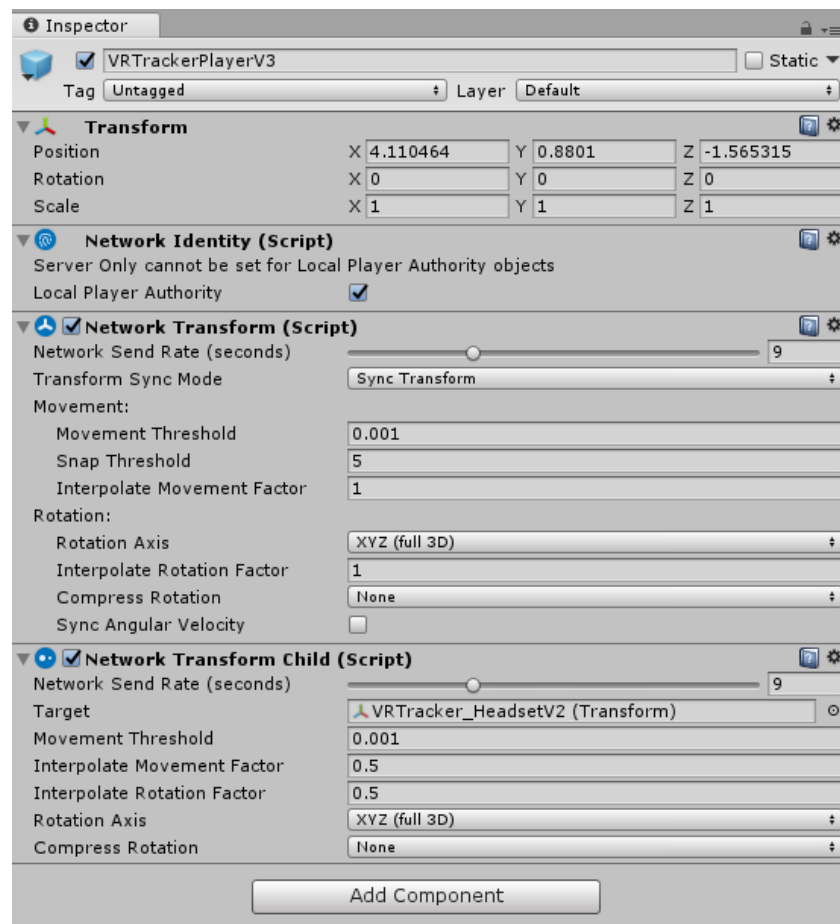
After adding you own scene in the Build Settings, add it as the Online Scene in the *Network Manager* of the **VRTracker** *GameObject*, this will be the scene loaded after the tag association.

## 5. The Player Prefab

We have left a prefab inside it so that you can see what are the different important element on it while using UNET.

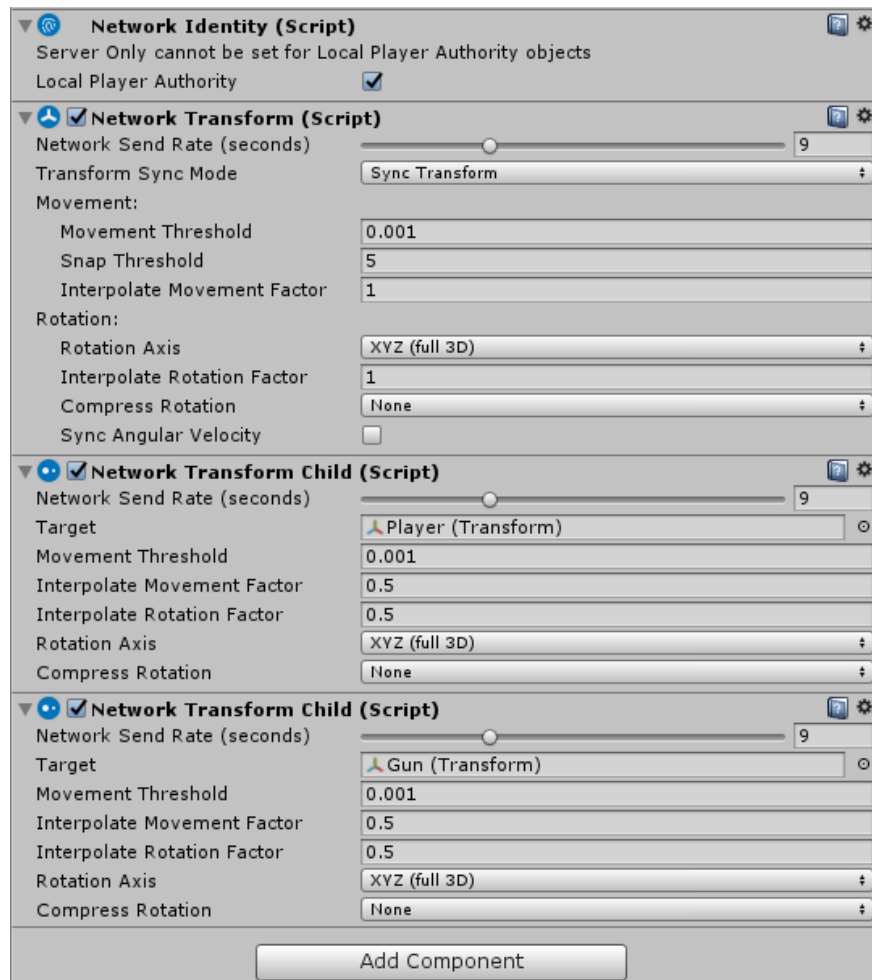


4. Player Prefab example



5. Components needed for UNET in the player prefab

You can see the different element here that are important for UNET. You can directly use this prefab to test it with your own scene or create your own player prefab. For that take your player and add on it the following component



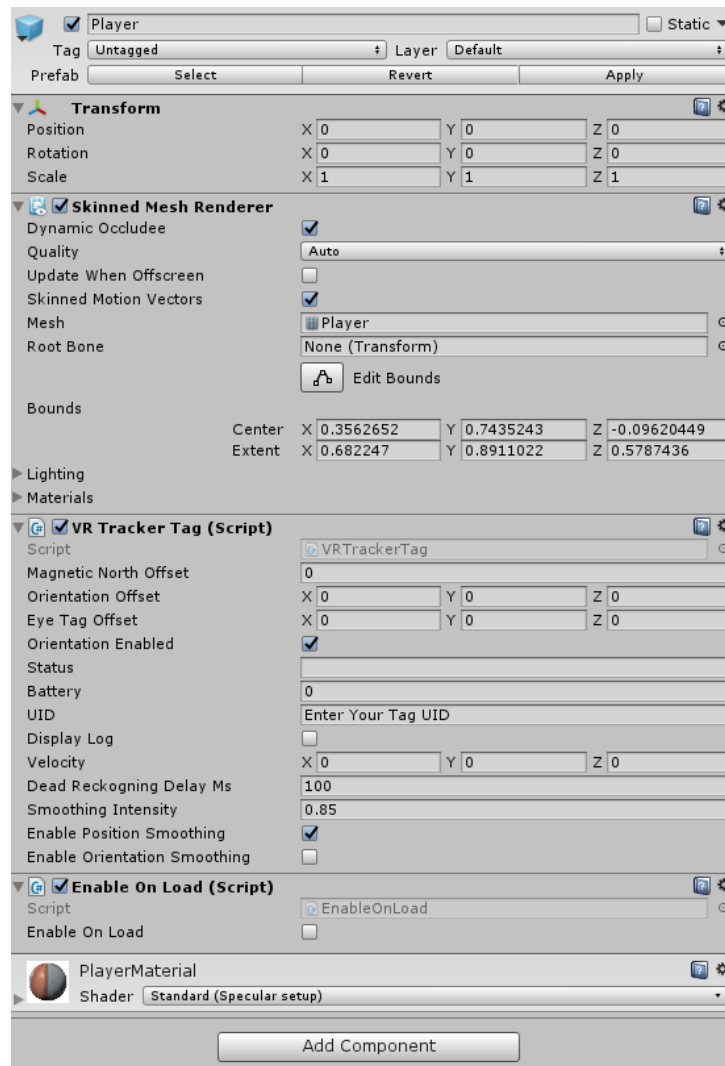
*6.Example of custom player prefab*

You will need to add the following components on the parent *GameObject* of your player:

- Network Identity, with Local Player Authority checked
- Network Transform, that will help network the player position across the network
- Network child, as many as the number of tag that a player will have. On our example with the Survival Shooter project, we need one tag for the player movement, and one tag for the gun. So, In this case, we need to add 2 Network transform Child, one for the player and one for the gun.

*(Do you think I should add the link with the description of the different element from Unity ?)*

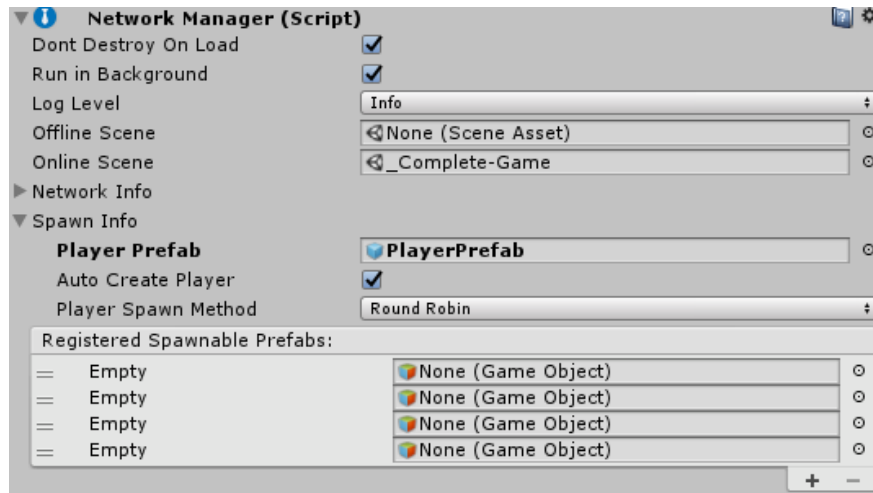
Now on each element that you want to be tracked, you will need to add the VR Tracker Tag script



In our case, you will need to add the script to the **Player GameObject** and the **Gun GameObject**. *Don't forget to add the camera to the player component.* And on the game object that will contain the camera, add the *Enable On Load* script, this script will enable the player camera when the scene will be loaded.



Once we have created the player, we need to link the player prefab to the network manager.



On the Spawn Info, select the player prefab that you have created as the **Player Prefab**. It will generate the different player avatar based on this prefab when connected.

Now you can try the game with the new user prefab that you've created 😊.

*(Would you like to get access to the code with the unity project ?)*