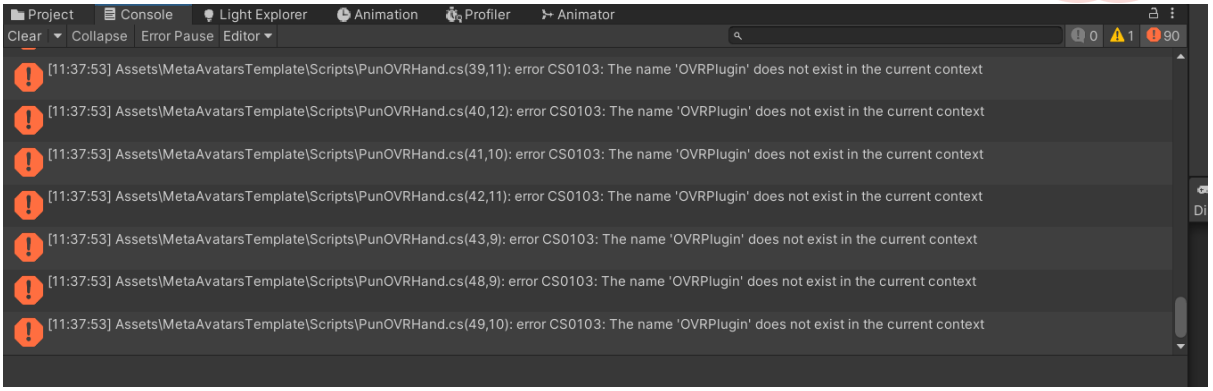
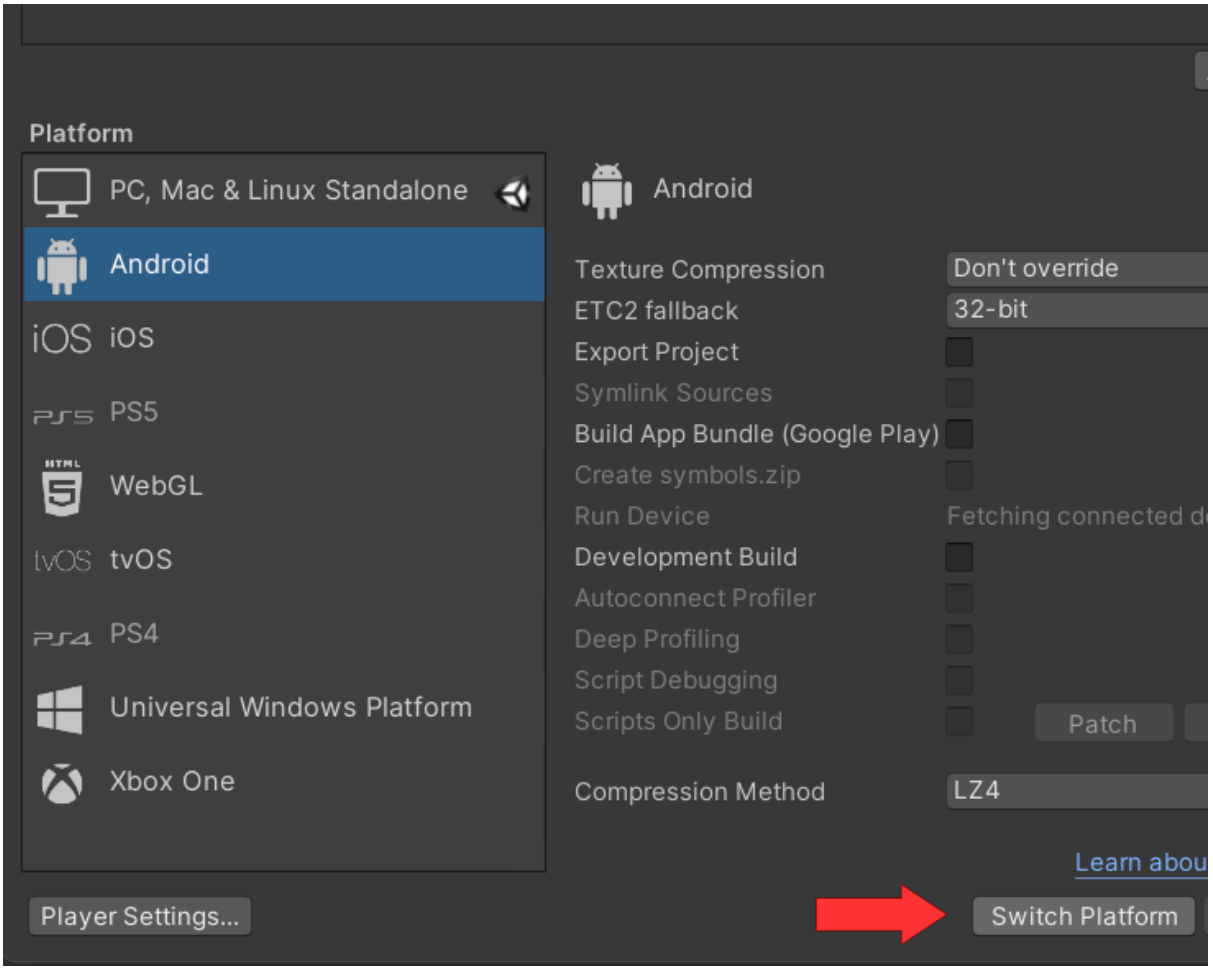
Meta Avatars Template setup guide.

After importing the package to a fresh Unity 2020.3.X project, you will see many errors. Don't panic! That's because we still need to import some other packages for it to work.



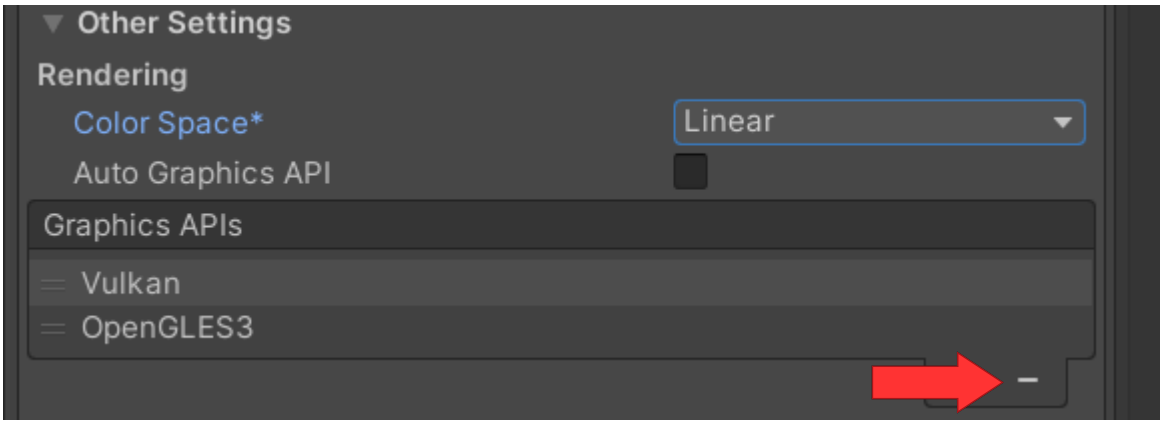
Let's start setting up Project and Build Settings.

1. Go to build settings and **switch platform to Android.**

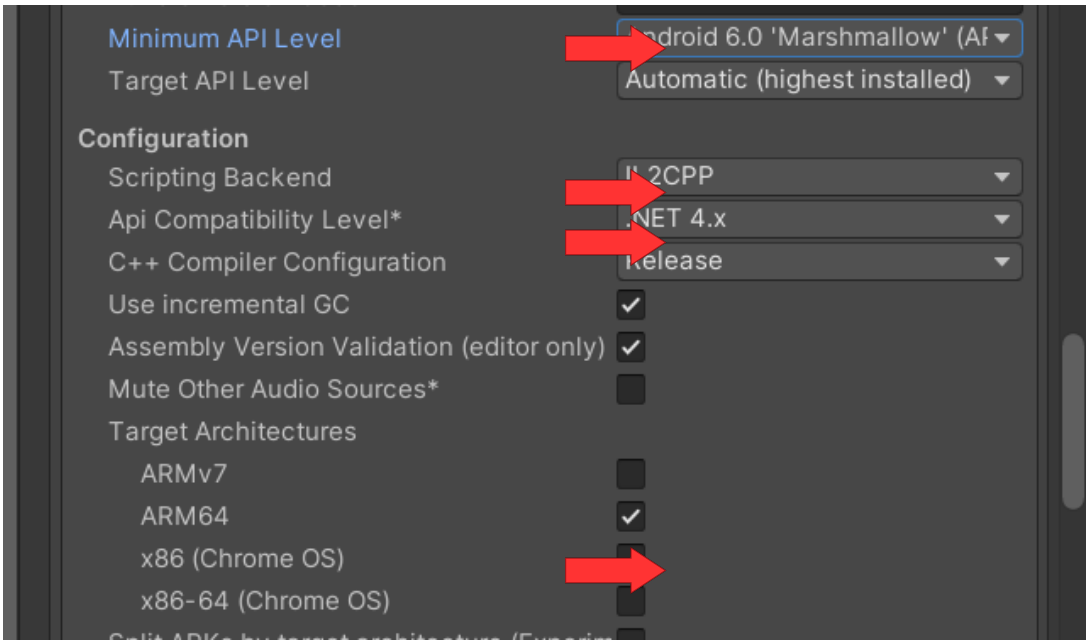


2. Let’s go to Project Settings > Player > Other Settings:

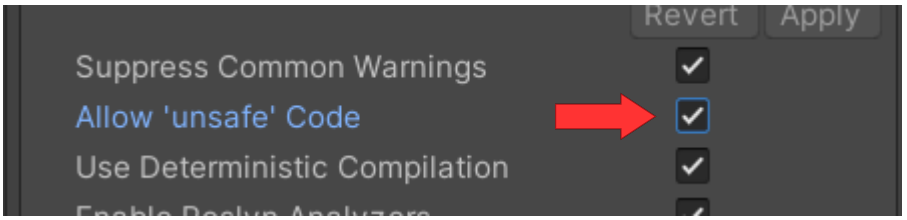
   Change Color Space to “Linear”, remove Vulkan from Graphic API list:



3. Set minimum API Level to 6.0 (API level 23), switch the scripting backend to IL2CPP, Api Compatibility Level to .NET 4.x, and set the target architecture to 64bits.

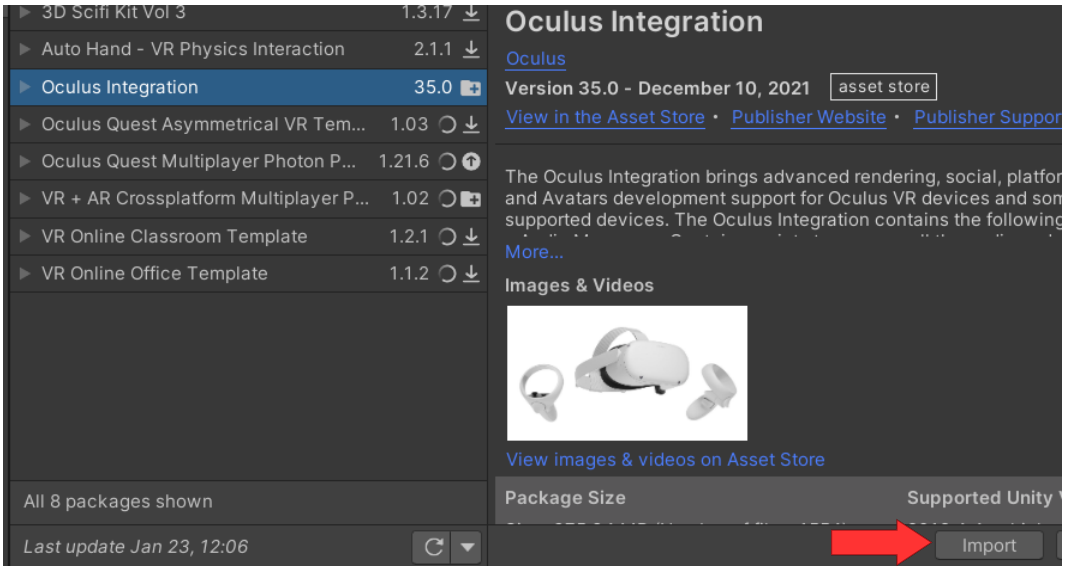


   - Enable “Allow unsafe code”



4. **Now, let's import some packages:**

- Oculus Integration v35.0 or higher:



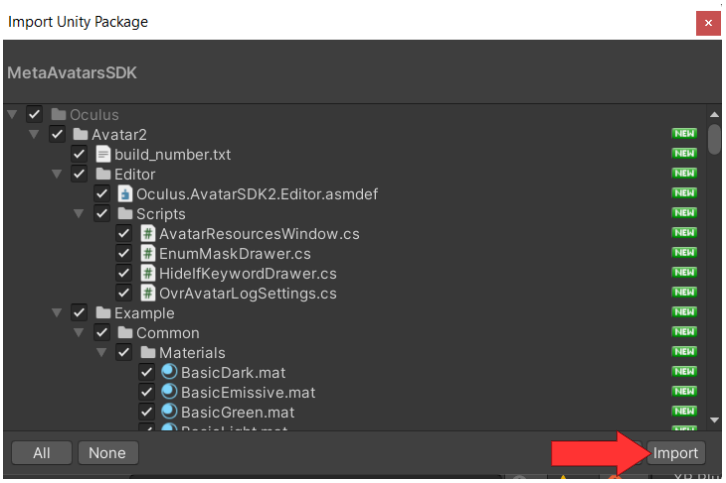
- Photon Pun 2 - FREE

- Photon Voice 2

- Go to https://developer.oculus.com/downloads/package/meta-avatars-sdk/ and download the Meta Avatars SDK (min version 9.1).

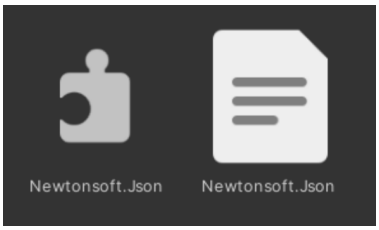
-

- **Import the MetaAvatarSDK** into your Unity Project.



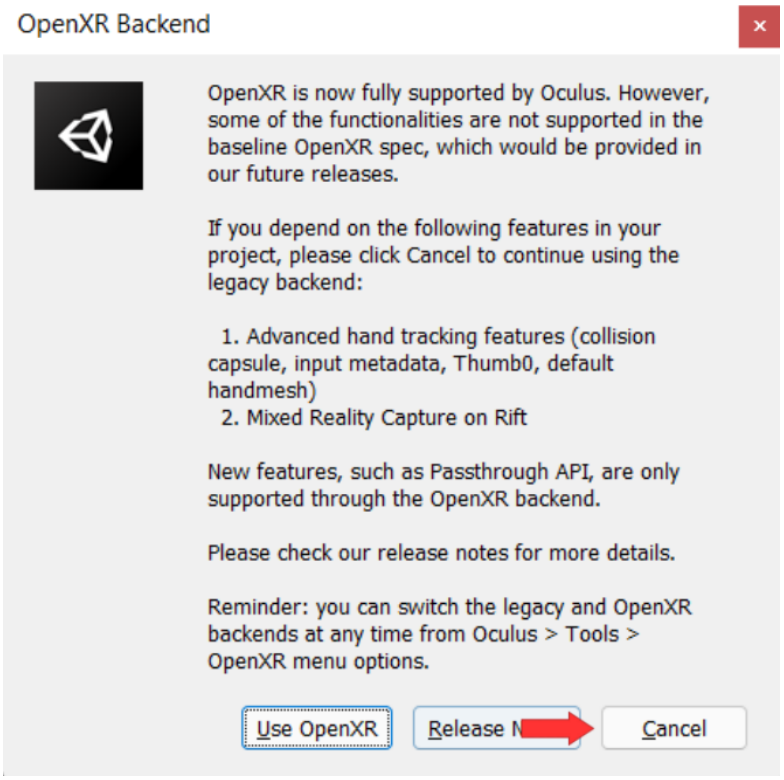
- If you get this error:

Go to Assets > Oculus > Avatar2 > Scripts > AvatarEdtorDeepLink, and delete these 2 files (they are duplicated).



- Now, the project should compile, and start asking for some updates. Accept all the updates requested.

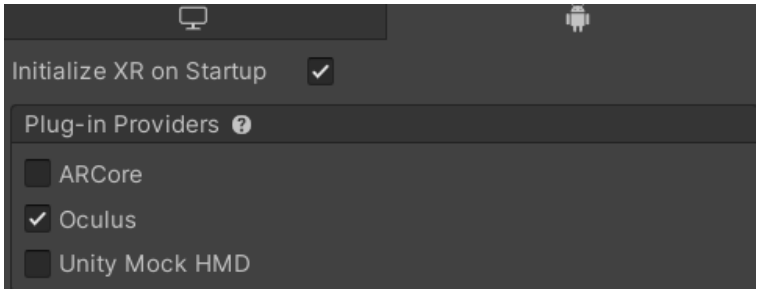
5. **Open XR** backend is optional. For hand tracking, it is recommended to stay with the legacy backend (clicking cancel). **You can freely switch later.**



- Unity will want to restart afterwards, click yes.

6. **Installing XR Plugin Management**
   Go to Project Settings > XR Plugin Management and install the plugin.

   Add the Oculus platform for both windows and android.

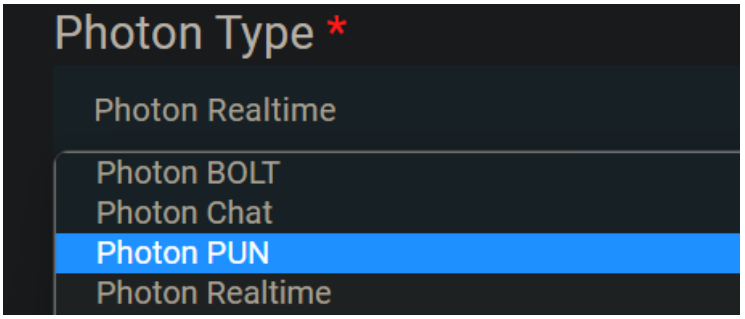


7. **Configuring Photon Pun2**

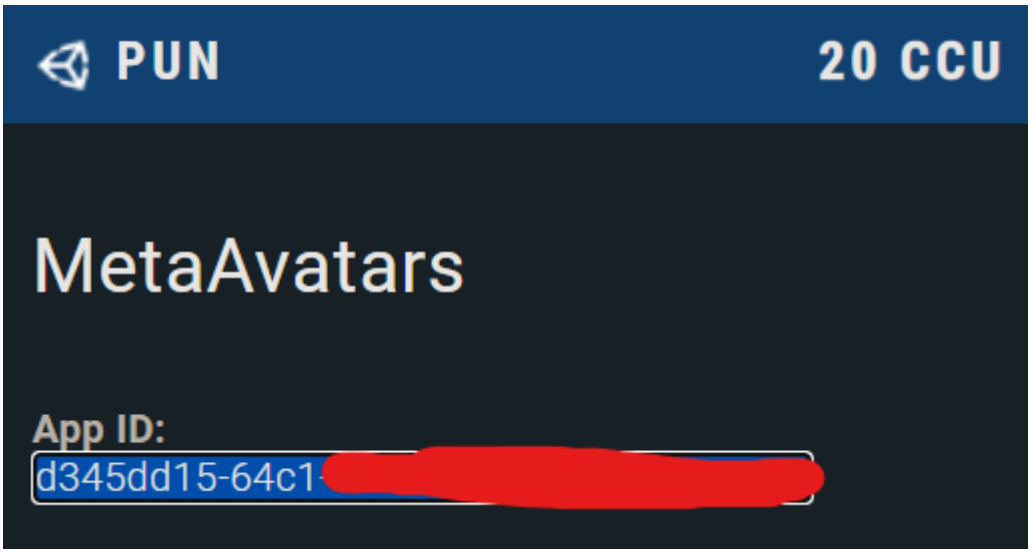
- Go to https://dashboard.photonengine.com/en-US/ and create an account if you don't have one.

- Create a new Pun App (you can create as much apps as you like)

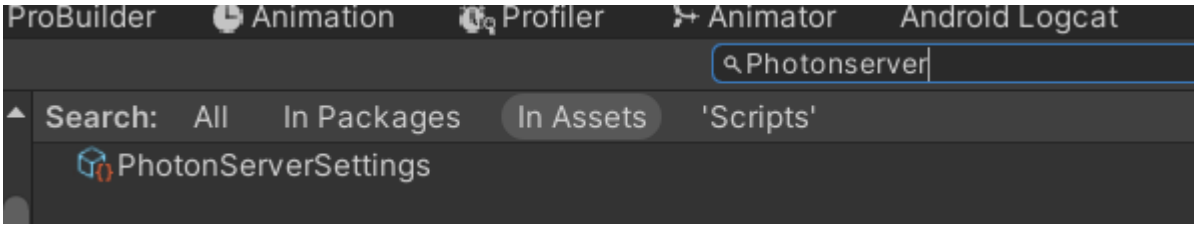
- **Make sure it's Type PUN**



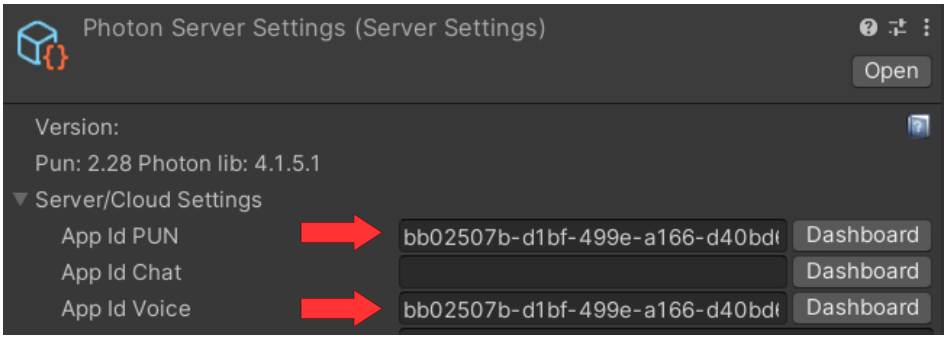
- Search for your created App in the dashboard, and copy the App ID



- Go back to Unity and look for the PhotonServerSettings configuration file in the project window.



- In the PhotonServerSettings configuration, paste the AppID in the PUN and Voice fields.

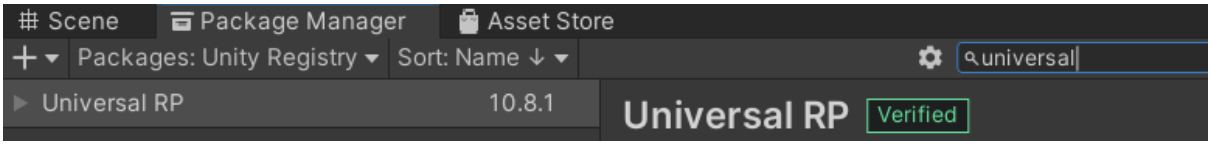


-

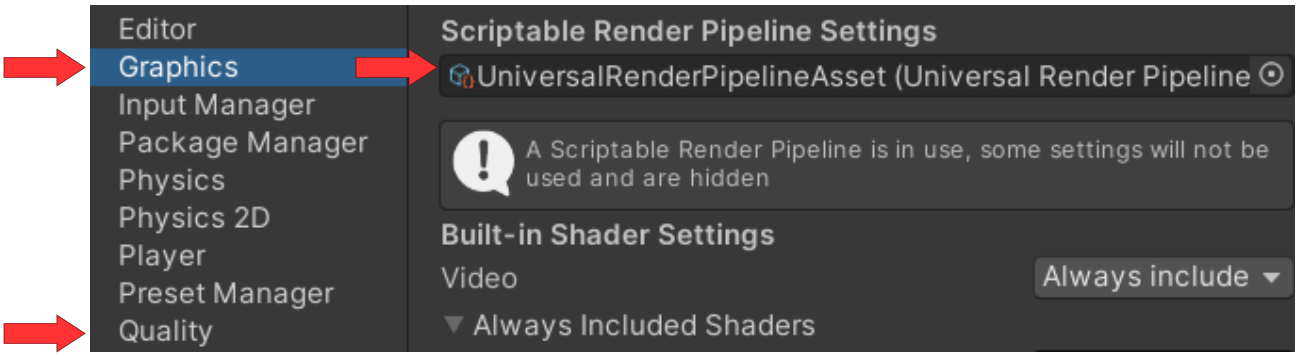
8. **Open the scene:** go to Assets > MetaAvatarsTemplate > Scenes and open MetaAvatar2Pun scene.

OPTIONAL: if you see everything pink like in this screenshot,

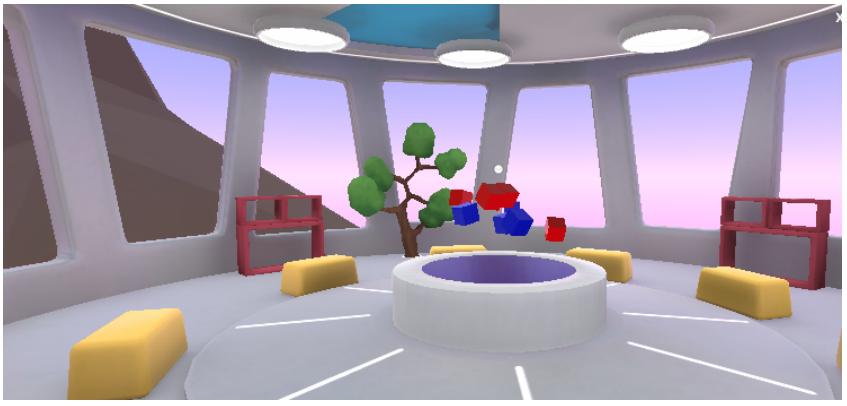
you need to install the URP (Universal render pipeline) package from the package manager (you find it under Unity Registry Tab).



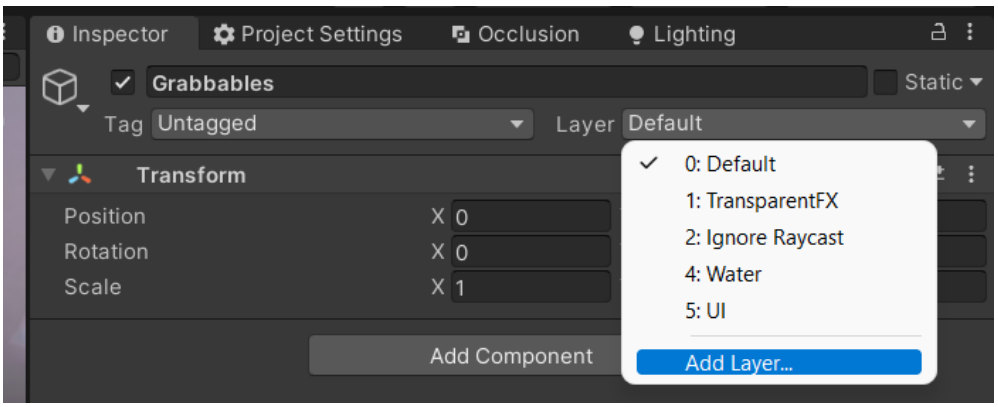
After importing the URP, go to Project Settings > Quality and Project Settings > Graphics and assign the UniversalRenderPipelineAsset into the fields.



Then, the scene should render appropiately.

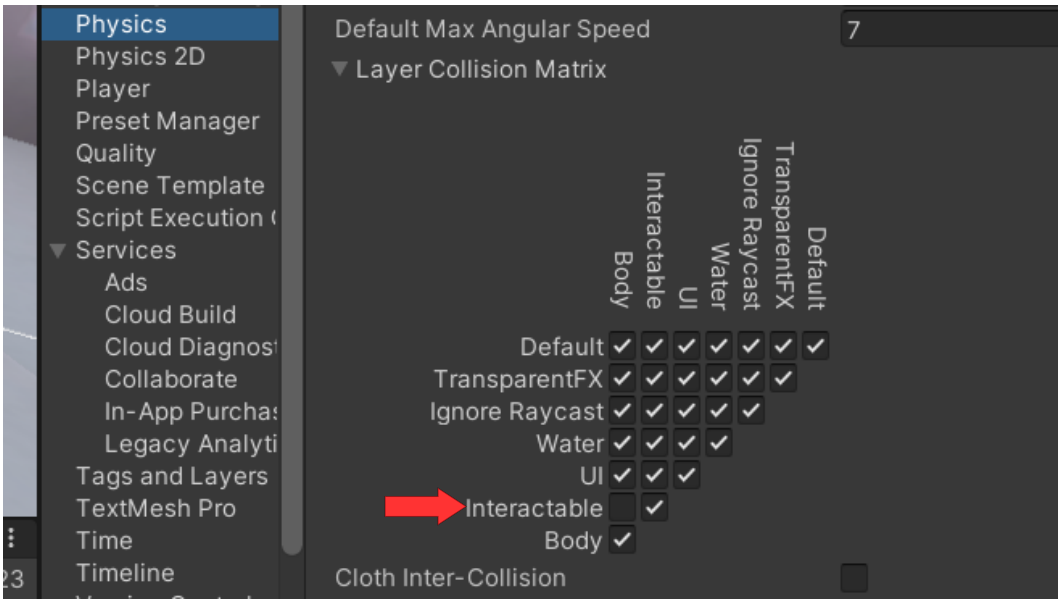


Setting up layers: the template needs a custom layer setup to avoid the body capsule colliding with interactables. To setup this, select any object in the scene and go to layer>add layer:

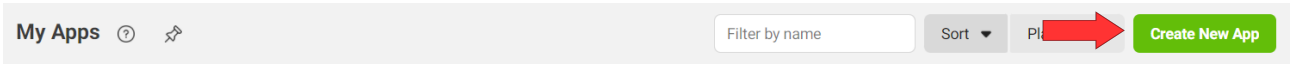


Then, click on the “presets” button in the top right:

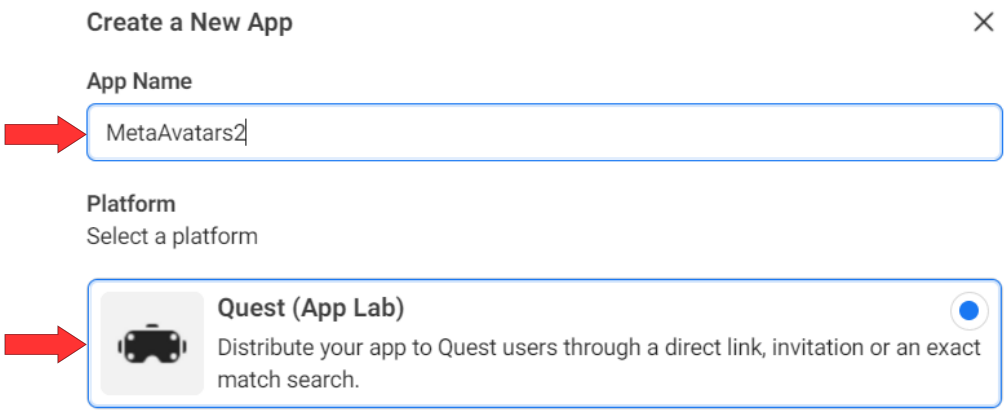
And select the preset included in the template.
Now, go to Project Settings>Physics, and make sure that the layers “Body” and “Interactable” doesn’t collide between them, unclicking it:



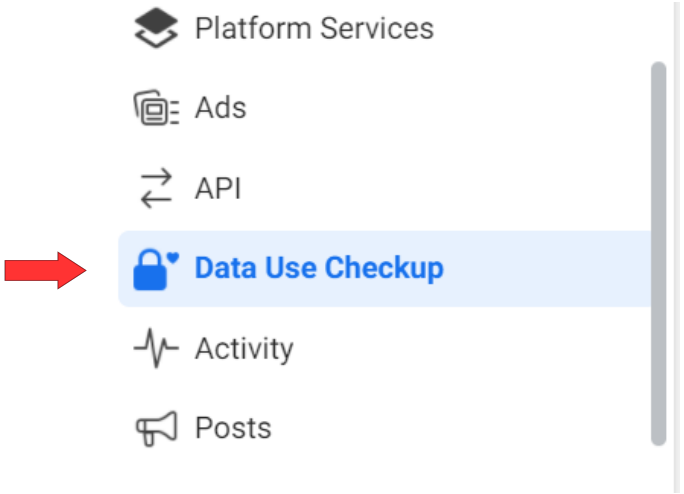
9. **Creating an Oculus App and enabling avatars.**
   In order to use the avatars, you need to enable them in the oculus developer portal. Please go to https://developer.oculus.com/manage/ and create a new app:



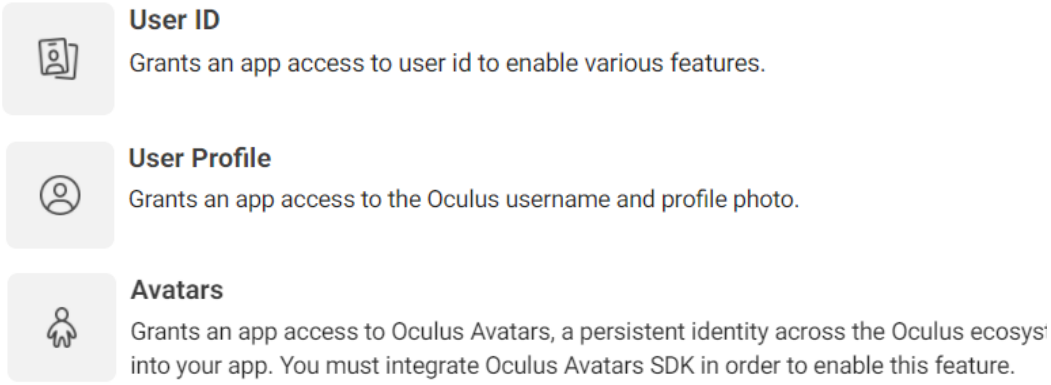
- Choose a name and select quest platform



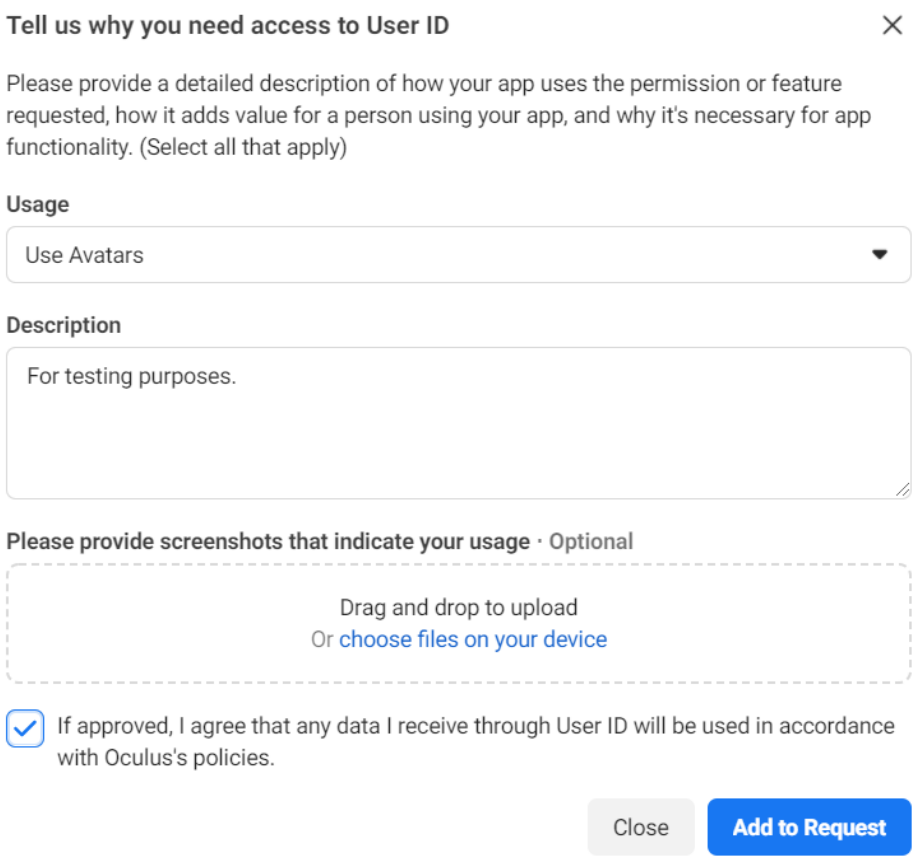
- Go to Data Use Checkup in the left menu.



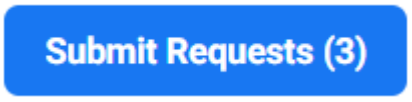
- Request access to the following features: UserID, User Profile and Avatars.



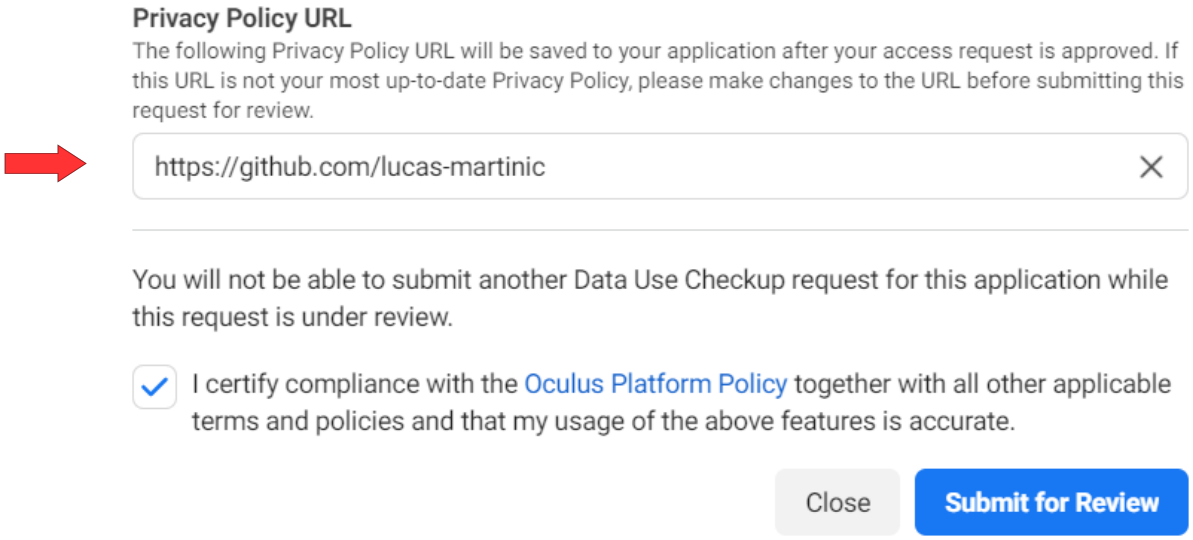
- Add the 3 requests like in this example, you can describe your personal purpose/use case.



- After adding the 3 requests, click on “Submit Requests (3)”

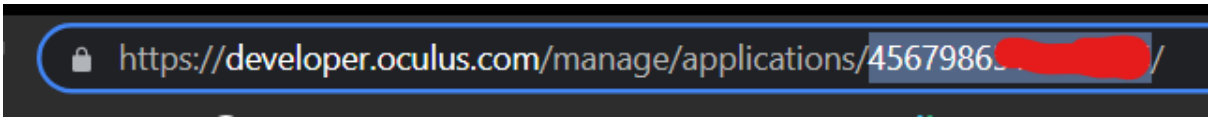


- You will be asked to provide a privacy policy, but as we will not yet be sending the app to the Oculus Store, you can provide a placeholder link, for example, your github account.

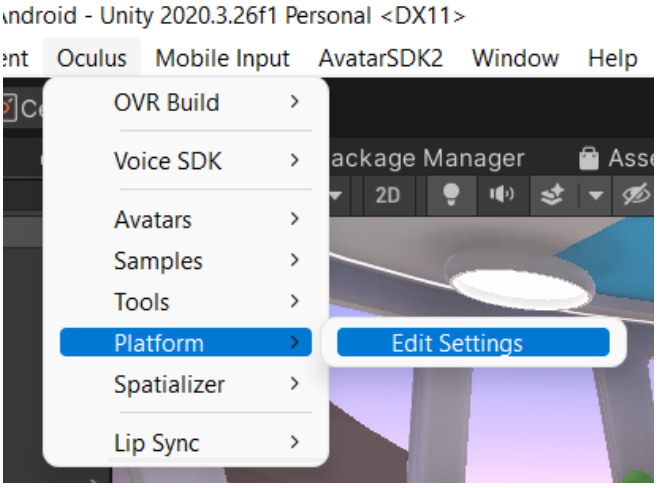


These features will be approved right away. Remember we need to complete this step because the Meta Avatars require an Oculus ID to work, as they use the user's personal avatar that is linked to their account.

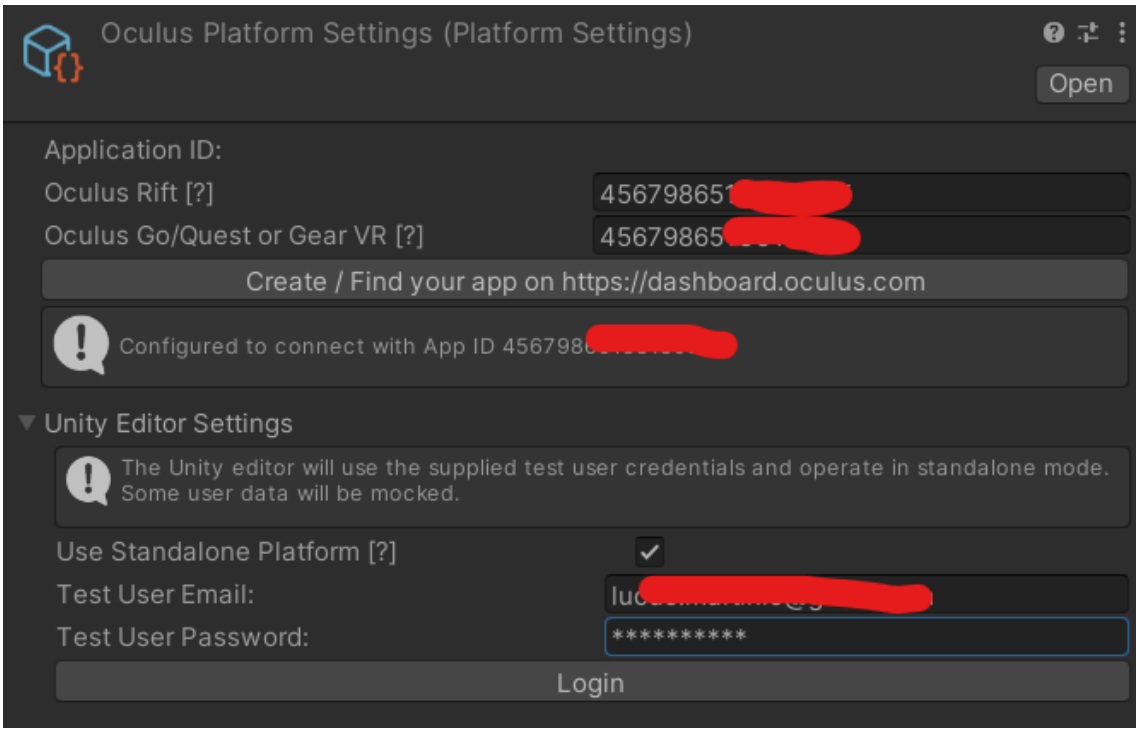
10. Now, copy your **Oculus App ID** from the top of your browser:



- And go back to Unity, and paste it in Oculus > Platform > Edit Settings in both Oculus Rift and Oculus Quest fields.



Also, select "Use Standalone Platform" and enter you oculus/meta credentials.

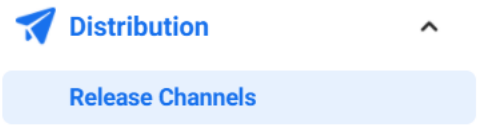


Now, you should be able to click on “Play” and the app should load your own Avatar linked to your oculus account!

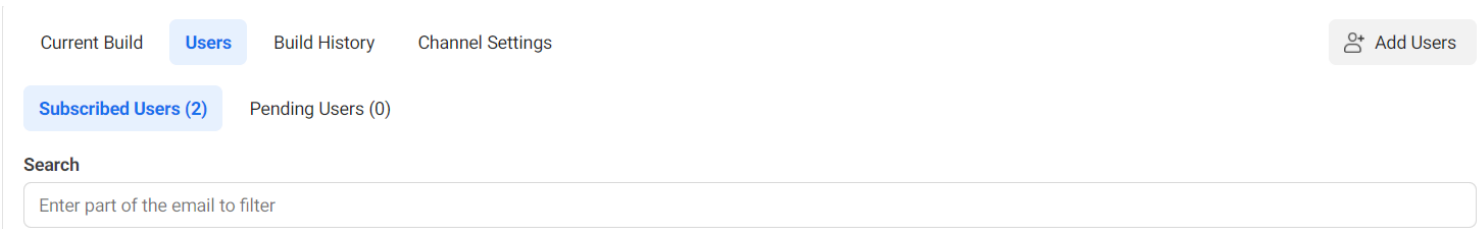
> Where’s the head?!? Don’t worry, this is a first person avatar (the one you see for yourself inside VR), so this is normal. Other avatars from other users will look normal, with a head :)

IMPORTANT

For an external player to join your app/game with their own Avatars, they must pass the “entitlement check” (normally, to own the game). While your app is still in development, you can go to the Oculus Developer Portal, and upload it to an Alpha/Beta channel.

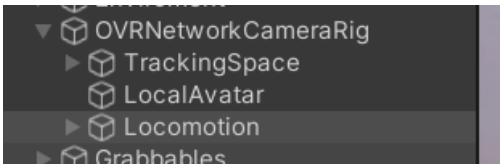


And then invite other accounts with their emails. Then they will be able to join and correctly load their avatars.

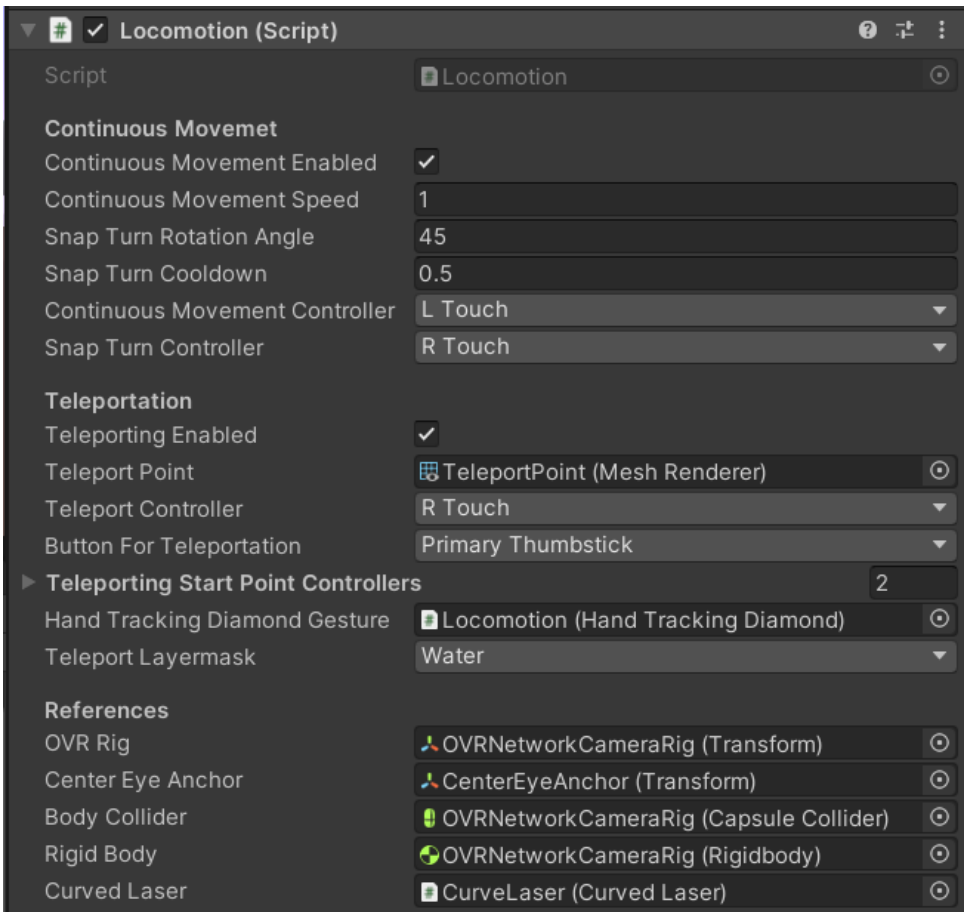


Locomotion System:

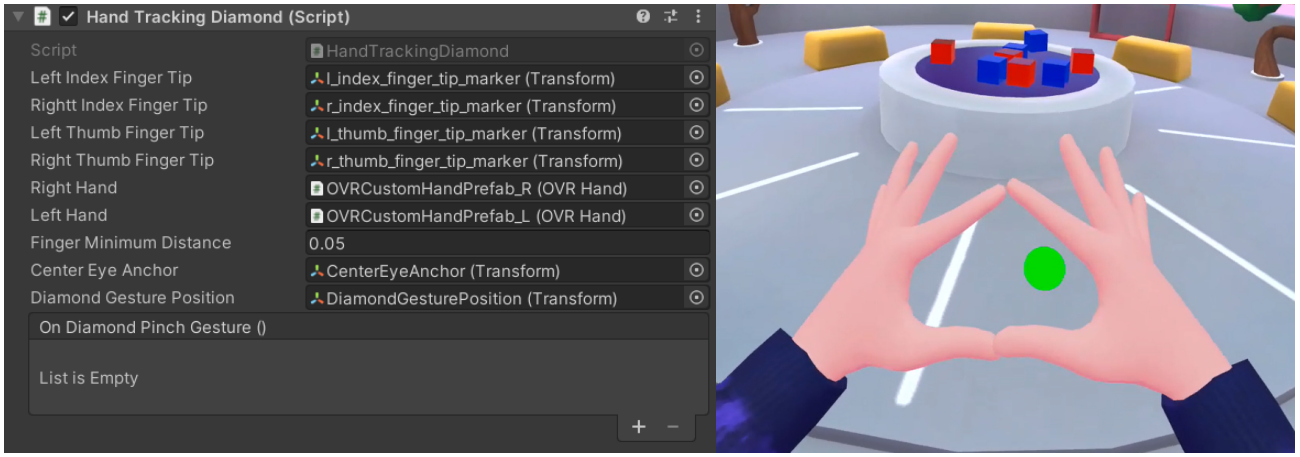
Since version 0.5, there's a custom Locomotion system included in the package, that will correctly work in a network.



Plenty of customization can be achieved through the Locomotion component:



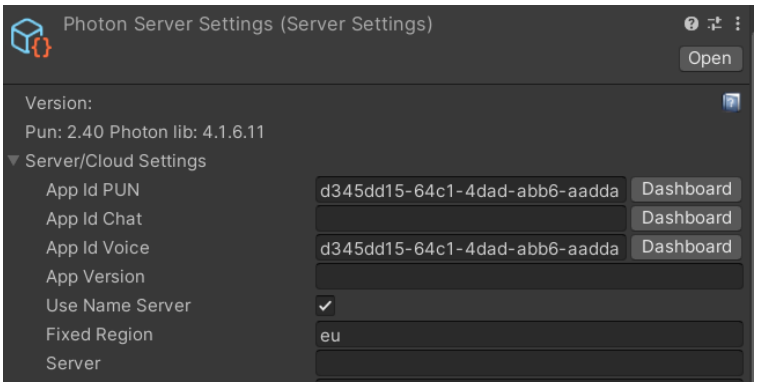
There is also a **HandTrackingDiamond** component which detects a gesture inspired in https://youtu.be/OkxyWVT0hoY?t=109 which is used as default for hand tracking teleportation. You can subscribe to the pinch event for custom behaviours.



Common Issues

1. “When I tested with my colleague, I am able to see my avatar and he is able to see his avatar, but we are not able to see each other's avatars.”

Please try fixing the Photon Region in the PhotonServer settings file, you can try with something like eu or us to force both clients to connect to the same Photon Server.



More info in Photon Regions here.