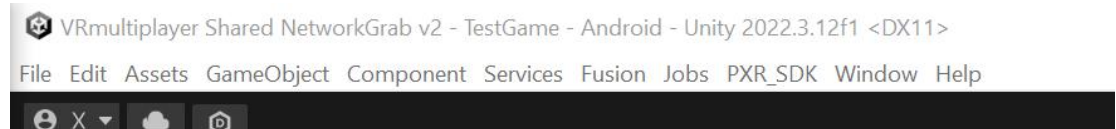


Setup Multiplayer

1. Open the VR template.

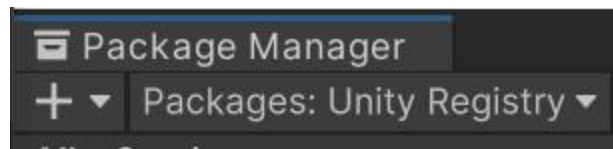
2. Check openxr plugin:

2.1. Go to toolbar that on the top of your unity editor,



Picture 1.1 toolbar

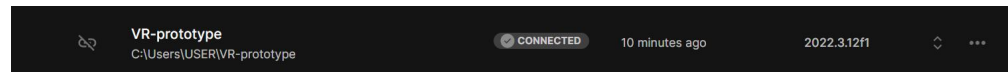
2.2. select Window, select package manager, select Packages: as Packages: Unity Registry



Picture 1.2 package manager

2.3. *note if openxr plugin version is not greater than 1.9.1, please update it by changing its manifest file.

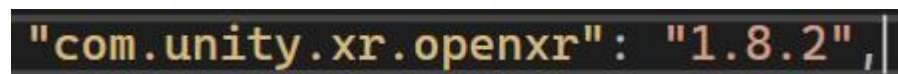
2.3.1. Close the project and go back to unity hub, right click this project or left click that 3 dots icon.



2.3.2. Select Show in Explorer

2.3.3. Enter your project -> Packages -> open manifest

2.3.4. ctrl + f to open search bar and type openxr



2.3.5. Change this from "1.8.2" to "1.9.1"

2.3.6. Remember to save the file

- 2.3.7. Reopen the file, the unity should show it is downloading packages.
- 2.3.8. Use same method to check openxr version



- 2.3.9. Now let's move on to the next task.

3. Set up Photon:

- 3.1. Create your own photon account, log in.
- 3.2. Download sdk, (this part can be skipped, since the project already have the sdk)

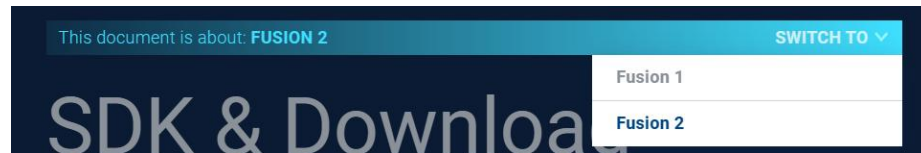
- 3.2.1. Go to the photon website and select SDKs at the top right side.



- 3.2.2. Select the product as fusion
- 3.2.3. Select the Engine | Framework | platform as unity
- 3.2.4. Click this button and click GET FUSION SDK button



- 3.2.5. This project is built with fusion1, so select switch to Fusion 1



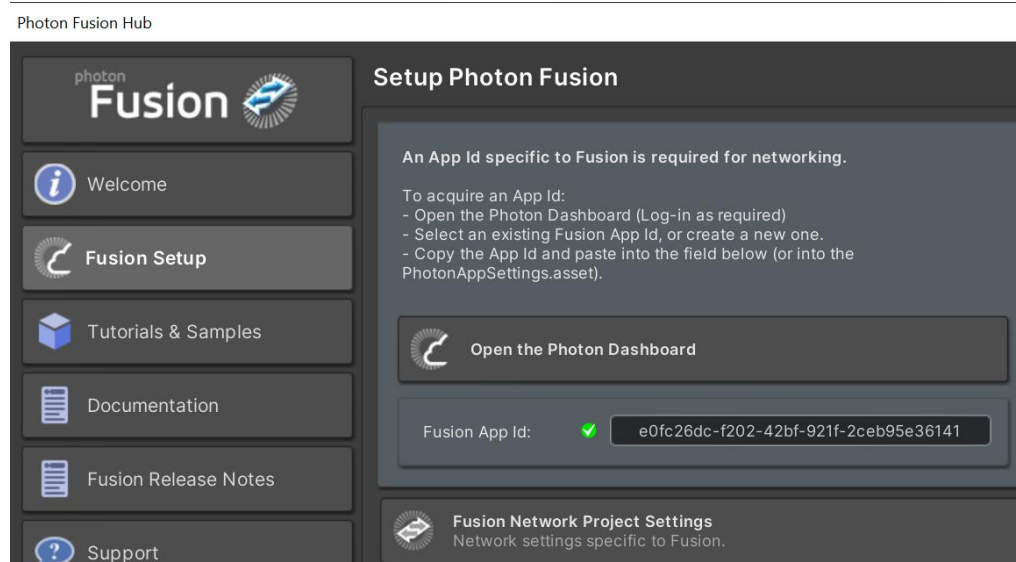
- 3.2.6. Download it from the link, the sdk should started with 1.x.x instead of 2.x.x

Version	Release Date	Download	
1.1.8 Stable	Jul 19, 2023	Fusion SDK 1.1.8 F Build 725	Release Notes

- 3.2.7. Go back to toolbar -> import package -> import custom package -> select the sdk

3.3. Create Photon game Id

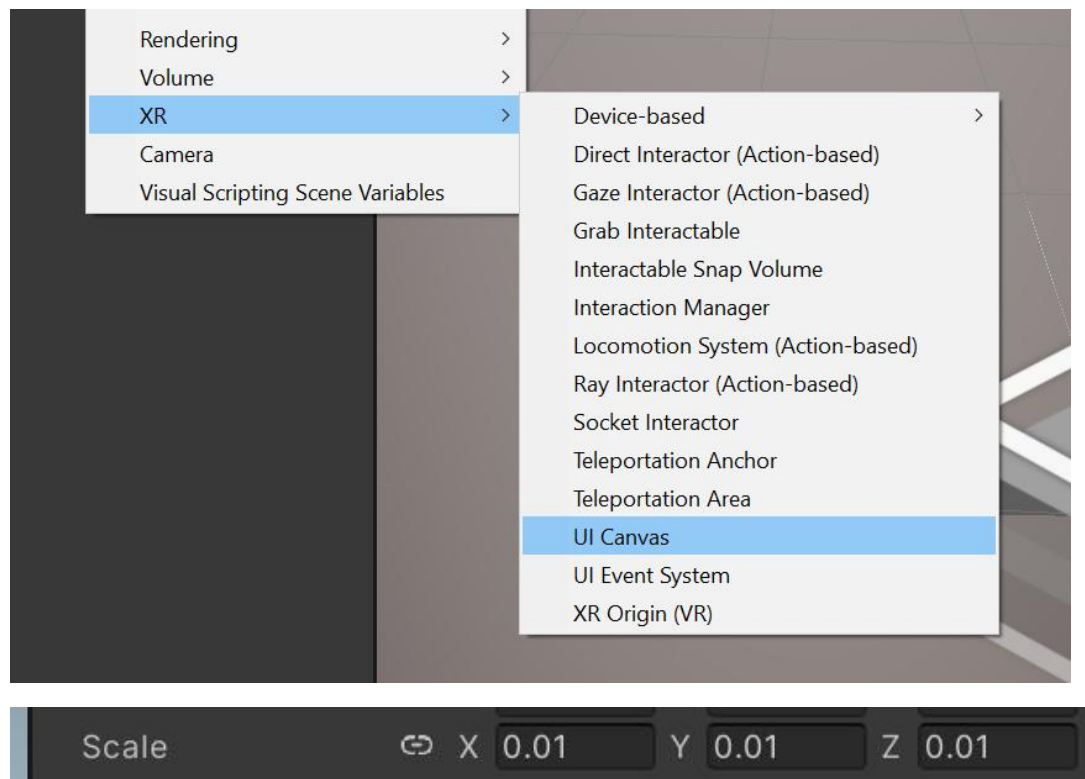
- 3.3.1. Tool bar -> fusion -> fusion setup -> fusion setup -> open the photon Dashboard.
- 3.3.2. Click create a new app
- 3.3.3. Select photon sdk as fusion, and select sdk version as fusion 1 and fill in other details
- 3.3.4. Copy paste the code into the Fusion App id and press enter



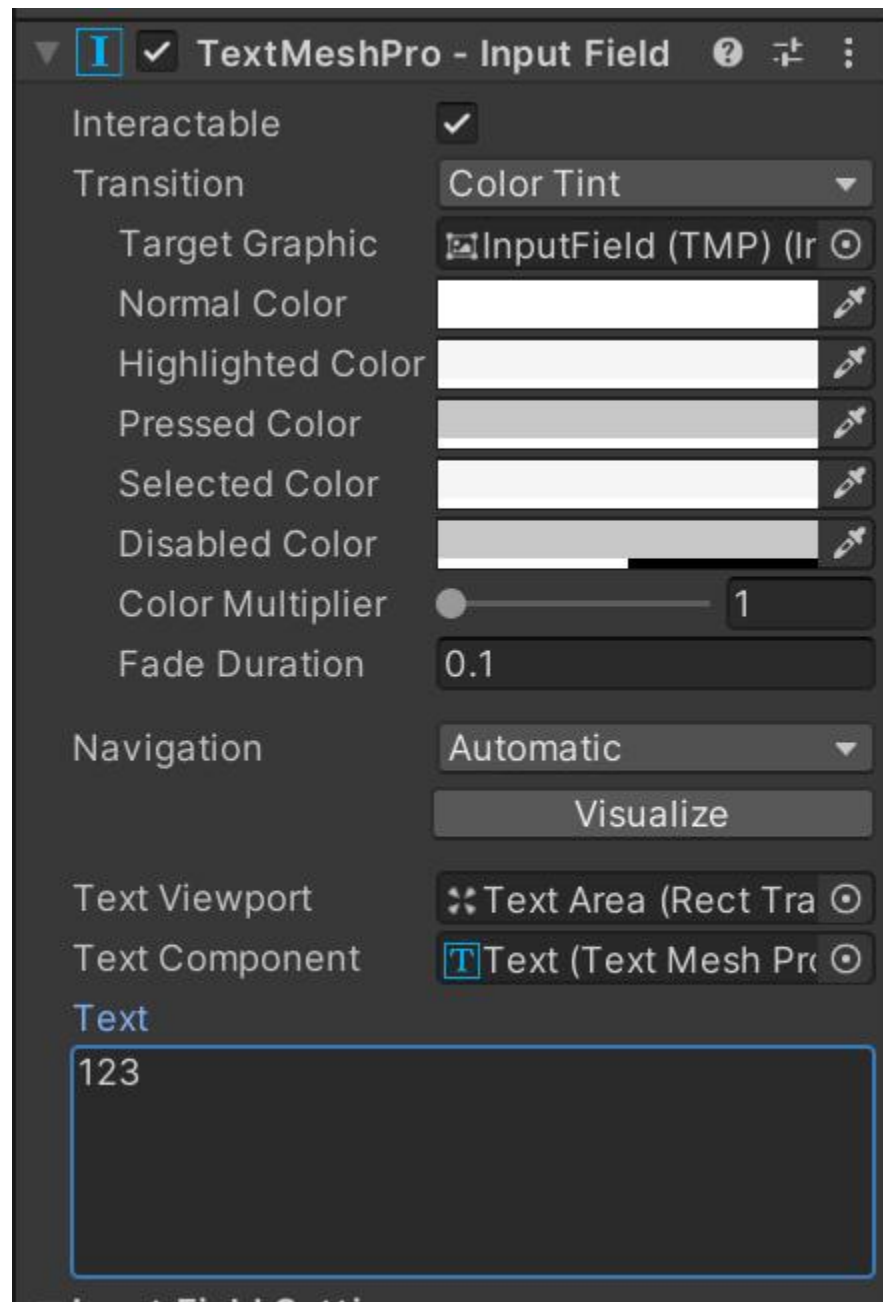
4. Set up hall (a place let user enter code to enter specific lobby)

4.1. Set up UI

- 4.1.1. Open the Prototype scene, create Ui by right click hierarchy, select XR -> UI Canvas, and scale it to 0.01

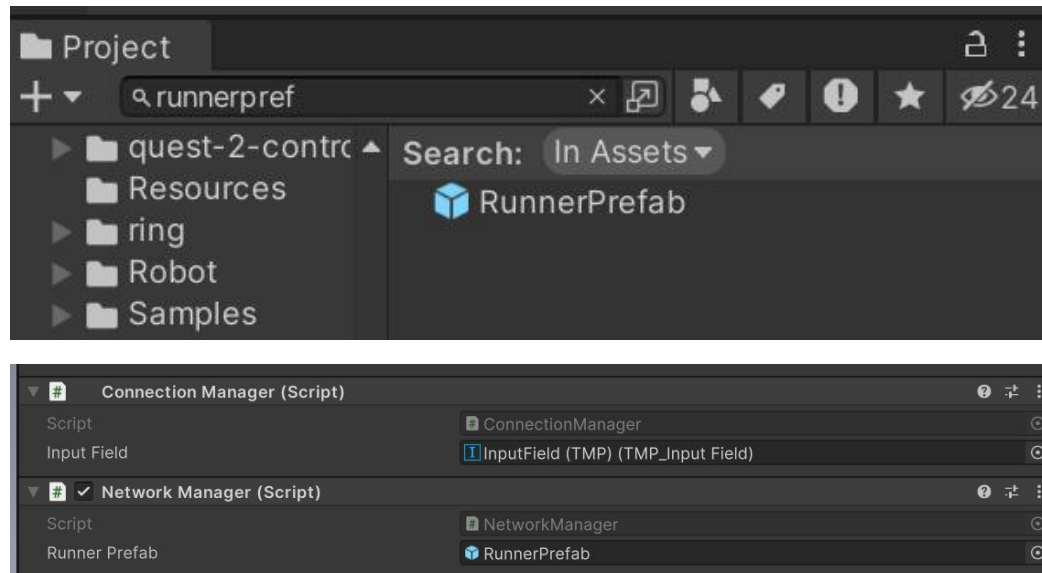


- 4.1.2. Right click Canvas and select ui -> panel
- 4.1.3. Right click panel and select ui -> inputField(TMP)
- 4.1.4. Select inputField(TMP), type room name into the Text, example 123. Thus, when another player also types 123 on this input field. After they click, join the room. They will join into the same lobby.



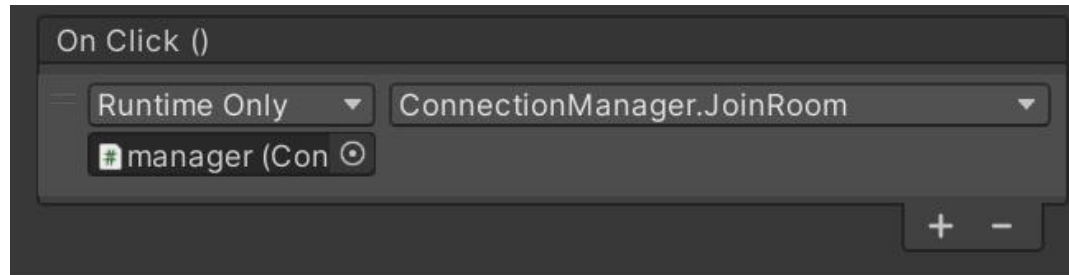
- 4.1.5. Can modify the width and height of the input field, select the text object by list down inputField(TMP), list down text area, select Text to adjust its font size.
- 4.1.6. Create join button for the user to enter the lobby, right click panel, select ui -> button- textmesh pro
- 4.1.7. Select the text inside the button object and change its text to "join the room", also adjust the size of the button like width:80 , height:20 and font size like 7.

- 4.2. Let's make them functional. Create an empty object called manager, and add **Connection Manager** and **Network Manager** code to it. Drag the input field to the connection manager. Search RunnerPrefab from the project search bar. Drag the RunnerPrefab into the Runner Prefab.



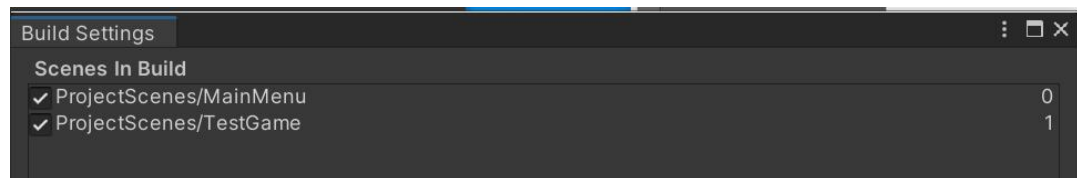
Explanation: the connection manager will pass the text to the network manager, Network manager will based on the code and search the room, if the room does exist, then the player will join the room, if the room does not exist, then the lobby will be created. Before joining the room, the game will create the network runner and then the player will load the scene locally, once the scene is ready, then join the game based on the setting.

- 4.2.1. Select the button that is inside the panel, and inside its Button component click the +, and drag the manager into it. And select connectionManager > JoinRoom



4.3. So a basic hall prototype is complete, lets user vr or XR Device simulator to test it.

4.3.1. Click file -> build settings, check the scene number. Current scene should be 0 and the next scene that will be loaded after player click the button is 1. (can refer back to the network manager codes)



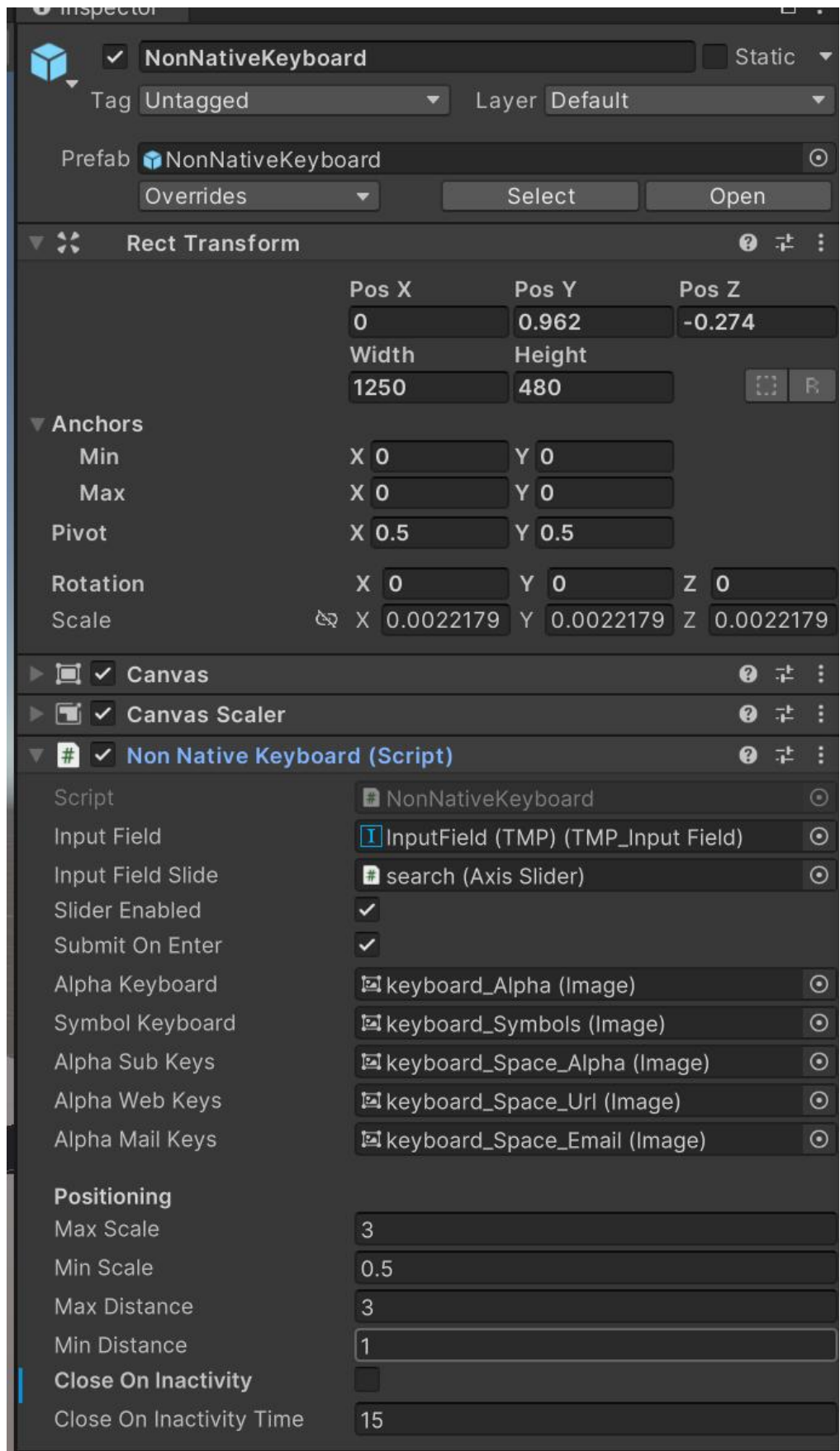
4.3.2. Play the game and test, *When tested with a VR headset please inactive the XR Device Simulator.

4.3.3. This scene 2 should looks like this

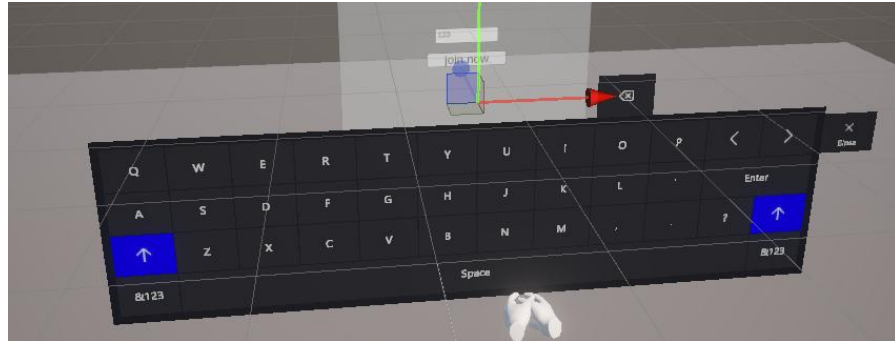
4.4. Let's add a virtual keyboard for the user

4.4.1. Search NonNativeKeyboard from the project files, put it into the scene and make it close to the panel.

4.4.2. Deactivate the close on Inactivity from the Non Native Keyboard components from Non Native Keyboard object



- 4.4.3. Remove graphic Raycaster from NonNativeKeyboard and replace it with Tracked device Graphic Raycaster
- 4.4.4. Back to the NonNativeKeyboard > keyboard_Background > search, disable background and inputField(TMP), and also disable the search's image components and Axis Slider components
- 4.4.5. The keyboard should look like

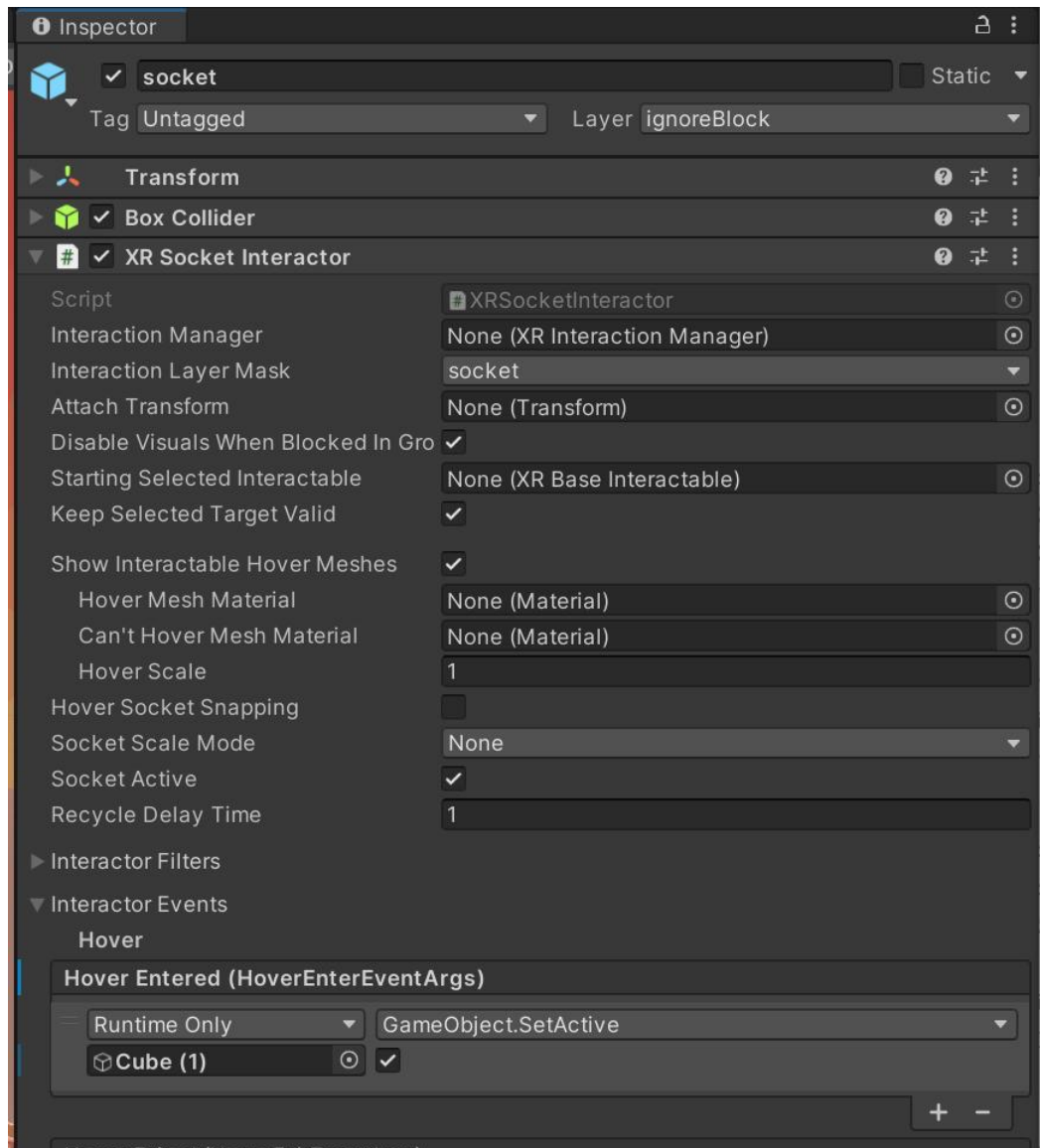


- 4.4.6. Back to the inputfield of the ui, add ShowKeyBoard script to it.
- 4.4.7. Run the game to test the keyboard is working or not
- 4.4.8. [How to Make a VR Keyboard in Unity - Poke and Ray Interaction \(youtube.com\)](#) can follow this video to improve the keyboard.

5. Set up lobby

- 5.1. Now we have to set up a basic electric plug in this scene.
 - 5.1.1. Drag the environment setup and place it on the floor while under the big red cube, feels free to adjust the cube as long as it is above the environment setup.
 - 5.1.2. Click the socket (Environment setup > Socket1 > handDetectionArea > socket), Drag n drop the Cube(1)

(Objective > Cube1) to the Hover Entered and event sets to GameObject.SetActive



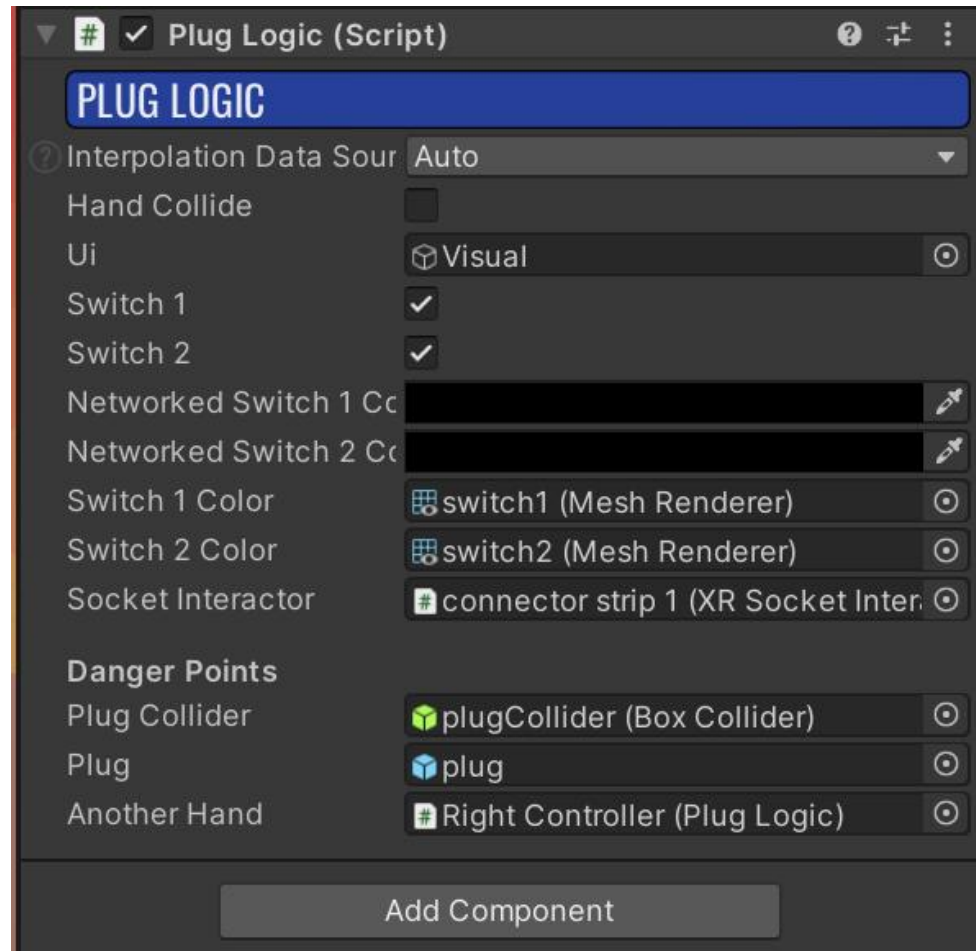
5.1.3. Before setup the rest of works, lets give a brief how the logic works in this electric plug.

5.1.4. **Explanation :** The logic will run on the hand, when the hand touches the switch, it will change switch1 or switch2 variable to true or false to indicate the switch is turning on or off. Before changing the variable, the

user will request a permission (stateAuthority) of the object, to let the server give the player the power to modify it. If they did not request stateAuthority, the switch will only be changed locally which other players cannot spot the changing. The color of the switch will be changed according to the variable which green = true while red = false. If one of the switches is off, then the player is allowed to grab the plug. Once the player puts the plug into the socket, it should display a red cube above on the extension. If both of the switches are turned on, when the player touches the plug, the player can see a yellow cube on its hand to represent the shock effect. (This effect only works on the first player that enter the lobby, while other player cannot see it on their hand, have not solved yet)

- 5.1.5. Open XR Origin > Camera Offset > Left Controller, add the PlugLogic script and Network Object component to it.
- 5.1.6. Drag and drop the Visual that under the shockEffect from the same hand
- 5.1.7. Drag and drop the switch1 (Environment setup > Plug Rig > switch1) and switch2 (Environment setup > Plug Rig > connector strip1 > switch2) to switch1 Color and Switch 2 Color.
- 5.1.8. Drag and drop the connector strip 1(Environment Setup > Plug Rig > connector strip 1) to the socket interactor.
- 5.1.9. Drag and drop the plugCollider (Environment Setup > Cable1 > plug > plug Collider)to the Plug Collider

- 5.1.10. Drag and drop the plug (Environment Setup > Cable1 > plug)to the Plug
- 5.1.11. Refer this picture setup plug Logic (this is the Plug Logic of Left Hand)



- 5.1.12. After everything is correct except another hand, let's copy the Plug Logic components by right clicking it and selecting copy components and paste it into another hand,
- 5.1.13. Remember to change the Visual of Plug Logic from another hand.
- 5.1.14. Remember to add the network object to the hand

5.1.15. Also attach plug logic from another hand by click the target button at the right side and select plug logic from another hand.

5.1.16. Let's test it.

6. Test the result with 1 pc with a standalone vr headset (recommended) or 1 pc.

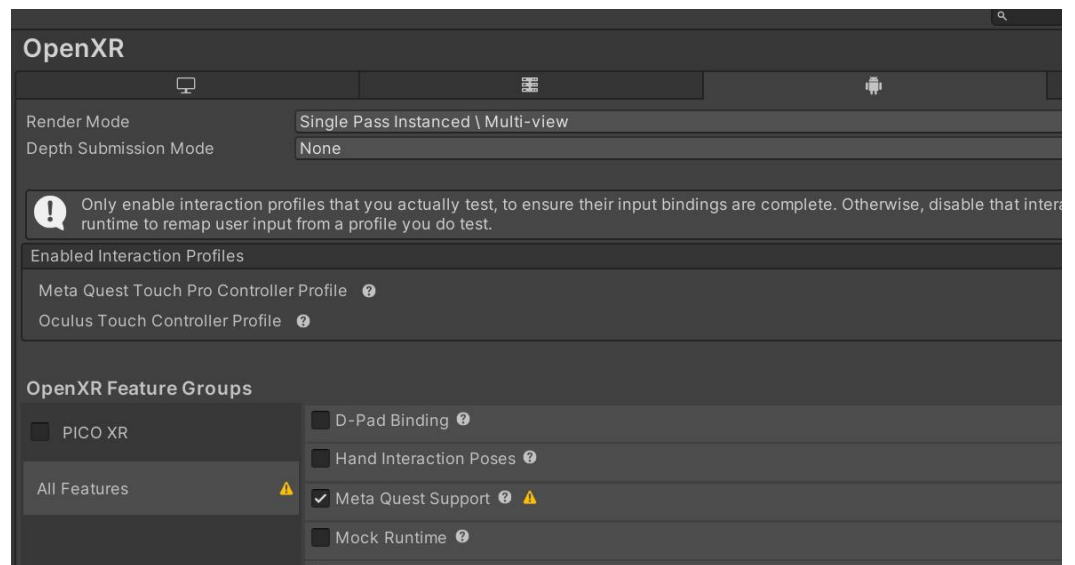
6.1. 1 pc and standalone vr headset (oculus)

6.2. Let's build a apk file first,

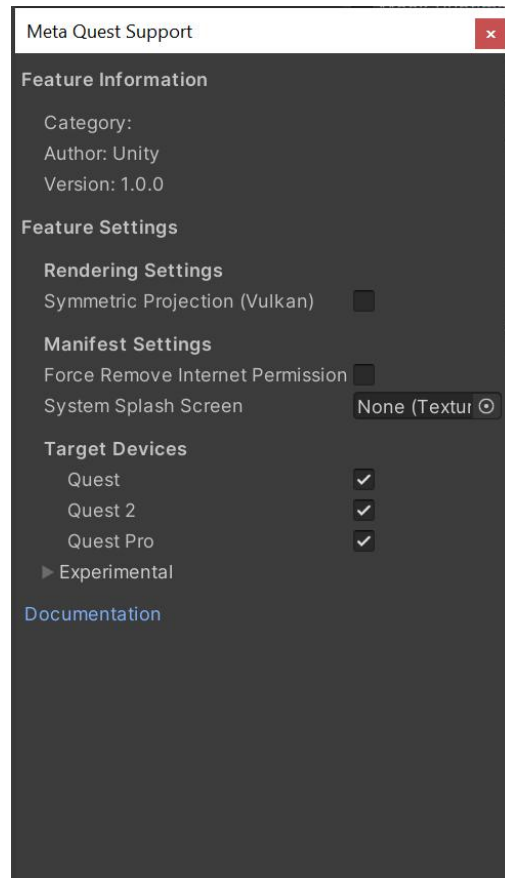
6.2.1. Before building the project, make sure the XR Simulator is off for both scenes, this gameObject will conflict with the VR headset user.

6.2.2. Edit > project settings > XR Plug-in Management > OpenXr

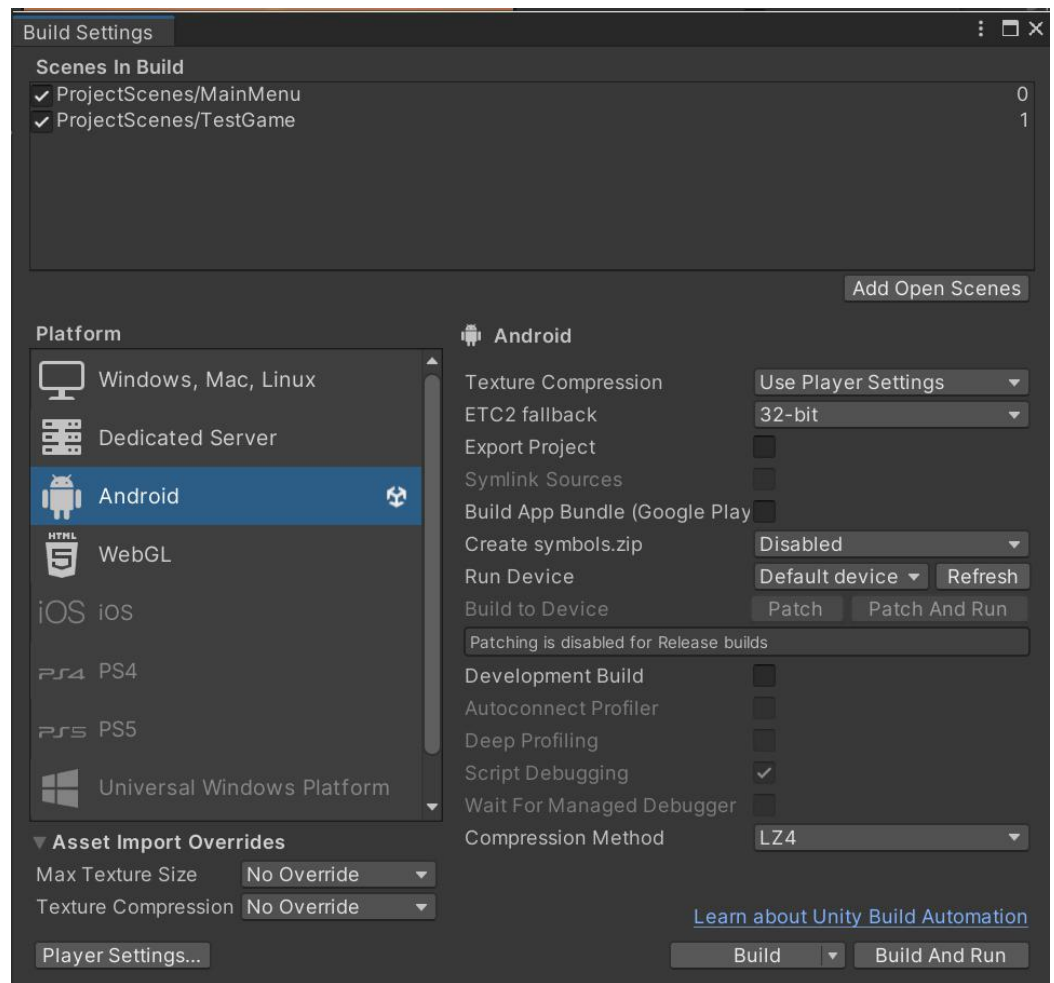
6.2.3. Remember to add oculus controller, and also the project is android not pc



6.2.4. Enable the Meta Quest Support in All features, also click its settings and make sure Force Remove Internet Permission is off



- 6.2.5. File > build settings and make sure the MainMenu and TestGame scene is in correct number and being enable, and platform is Android with these settings



6.2.6. Click build to build the project.

6.3. Test

6.3.1. Use SideQuest to load the project to your headset.

6.3.2. Run the project on oculus quest 2, and use another headset to connect to the pc and run it.

6.3.3. Make sure both of them enter the same lobby id.

6.3.4. After switching to the next scene, they should be able to see each other after hand models have been spawned.

6.3.5. Test it with these test plan

Test plan:

1) Yellow cube can u see it?

2) Switch button will change color correctly from both side

Color will update correctly after different hand touching example: left hand touch upper switch, right hand touch lower switch, they will update correctly, the switch1 and switch 2 variable should be synchronize from both hand.

3) Plug grabbing also works correctly (only can be grab when one of the switch is off)

4) Both of the player can see Red box when plug is put into the socket.

6.3.6.

7. Extra (learn)

1.3.1 Multiplayer (Basic)

3 type of multiplayer mode in photon fusion

Host-client mode, shared mode, dedicated server(server acts as a host)

Pros & cons (what i experience)

Host-client mode can deliver better physic interaction in vr, your hand can be block by table.

Shared mode cannot achieve it but it is still able to let the user grab and detect triggers in vr. The object that is held by the player can do some collision with an object that is not grabbed by another player.

How photon fusion shared mode works:

Everyone is in single player mode at the start (can be skip),

After that, when they join the rooms, the game will become multiplayer.

The player only can affect the game object and the variable when they have the permission (stateAuthority <each network object has their own stateAuthority>), if they don't have the stateAuthority of the game object, it will only work on local, the gameobject will not be affected and sync to other player's game. Photons

fusion only collects data from the player that has state authority and sends it to other players.

We just has to add the function that prepare by photon

```
await Object.WaitForStateAuthority();
```

This code will request itself stateAuthority and transfer to the player. This is a await function means it will request without blocking other code running, thus only the code below will have to wait for this code to get stateAuthority. The rest will just run without being block.

In shared mode, everyone has their body's state authority, but other script object will be own by the first entered user, unless

Inside the example,

This project is built by photon fusion 1 in shared mode.

Inside shared mode there is 2 type of grabbing solution

Local grab or network grab,

Local grab: cannot prevent the player from grabbing the object by just unticking a grabbable object.

Network grab: can prevent the player to grab the object by just untick the network grabbable object

[\(11\) How To Make a Multiplayer Game in VR? - Photon - YouTube](#)

Follow this video to create a simple VR multiplayer

* create UI with xr -> ui

* if want develop openXR multiplayer, u should not using oculus integration bla bla

* standalone vr support can be solved by install latest openXR version which is 1.9.1 and then untick the force remove internet permission setting. Save it and restart it and ensure it is untick.

* A issue: the hand reset to default rotation, (solution create a parent, and set it to your interpolation target instead of visual hand) (this issue also can be solve and be simplifier by upgrading to fusion 2, but consider the experience i had with fusion 1, I suggest only upgrade to fusion 2 when u really need it.)

If u face tons of error in unity that relates to photon and unity that listed on (0,0)

- Dont put [Networked] on unity stuff, example: Render
- [Networked] variables should only be variables
- Inside the documentation, they encourage use networkbool instead of bool, haven't find out the reason yet

1.3.2 Multiplayer plug

High level concept of the plug:

When switch 1 (that one near the invisible wall) is on while switch 2 (that one at the extensions) is also on. Players cannot grab the plug, they only can grab the plug when either one of the switches is off. If the player wants to grab the plug, there should be some shock effect come out / ui warning to notify the player they cannot grab it now, it is dangerous. When they grab the plug and put it at the socket. There should be something to notify player they finish the mission (using red block in this example)

1.3.3 XR Socket Multiplayer Toggler

This script is written to build a bridge that connects between xr socket with photon fusion. The benefits of this script will be using the existing unity xr toolkit with photon without building a whole socket script for photon fusion.

Network variable only synchronize between user, cannot synchronize between

Ver1: (it can't synchronize the switch1, switch2 variable between local both hand)

Solution: have to separate the code, only 1 script that stores and updates the switch. And it should be put at an empty object.

Each

1.3.4 Improvement & bug

1. No hand animation: The example does not use the hand animation code that is provided from the photon. Basically change animation according to the hand command.
2. Socket Visual problem: At the end of the mission, the plug is being put at the socket, its rotation is not the same as the attached gameobject, instead it remains its 0,0,0 rotation.
3. Socket Visual problem: the plug sometimes looks vibrating at its start socket.
4. PlugLogic: the code contains trigger detection and update switch variable and make the plug grabbable after condition is met , the trigger detection should have separate script with the switch variable and plug grabbable.

5. The socket system is not fully completed yet, it is functionable to grab an object that is not inside the socket and put it at a socket. If the object is inside the socket at the start, the developer will have to write code to disable the socket and activate the network hand collider grabbable the object.

6. VR too high

7. Player wont disconnected

8. Shockz effect is not displayed?