

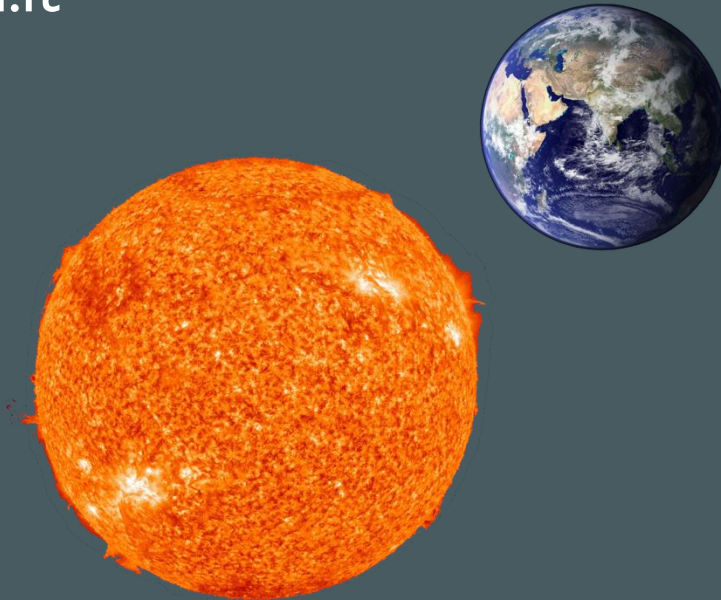
# SOLAR SYSTEM EXERCISE: HANDS ON TOGETHER

Corso Realtà Virtuale 2024/2025

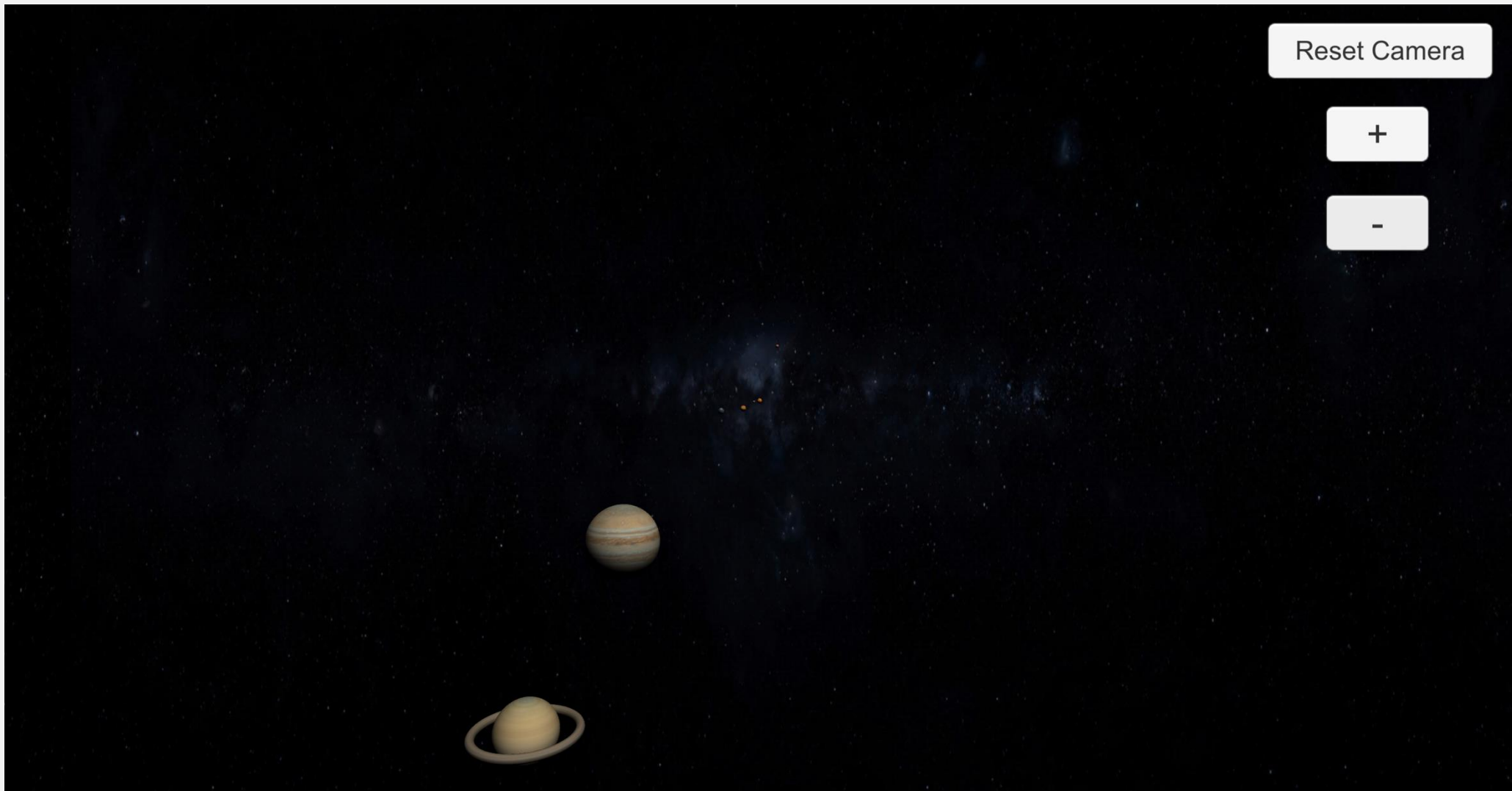
eleonora.chitti@unimi.it

Code on the laboratory repository Github in the folder  
Ex07-01

<https://github.com/aislabunimi/courses.vr2025>



# SCREENSHOT OF THE FINAL RESULT 😊



# UNITY3D WITH XR ARFOUNDATION



# GET STARTED WITH ARFOUNDATION

AR foundation allows to work with different AR platforms inside Unity3D

You can install ARFoundation in Unity's Package Manager

In addition to that Package, you should install also the Plugin you need, as ARKit for IOS apps or ARCore for Android apps development.

## Supported Platform Packages

The following platform packages and later implement the AR Foundation features indicated above:

Package Name	Version
ARCore XR Plugin	4.1
ARKit XR Plugin	4.1
ARKit Face Tracking	4.1
Magic Leap XR Plugin	6.0
Windows XR Plugin	5.0

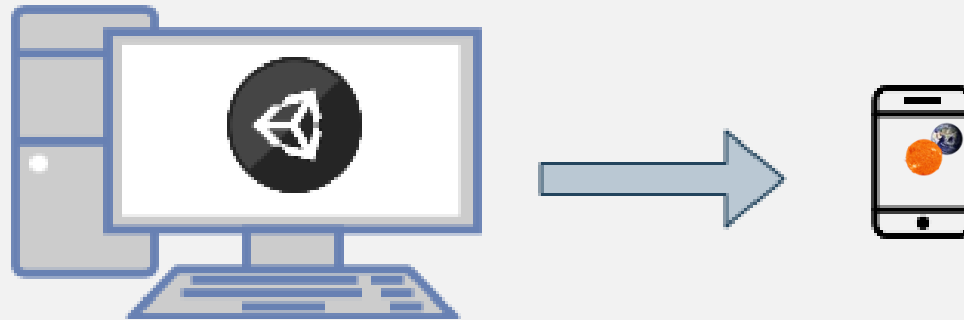


<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#using-ar-foundation>

# WHAT TO DO NOW

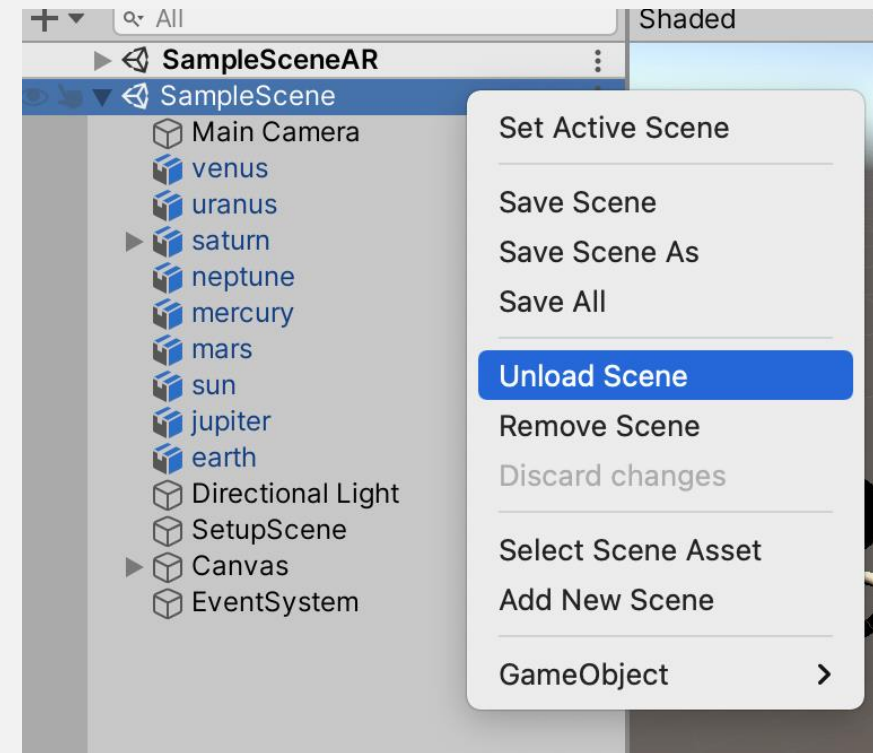
At this point we have modeled a Solar System in a Unity 3D Scene named *SampleScene*.

Now we will use the same assets to create the XR version, then we will build it for IOS with ARKit + ARFoundation



# CREATE A NEW SCENE

1. Create a new scene: File > New Scene , save it and call it *SampleSceneAR*
2. Go to Assets folder and right click on the *SampleScene* -> select Open Scene Additive
3. From the *SampleScene*, copy all the planets Game Objects and the SetupScene Game Object
4. Right click on the *SampleScene* -> select Unload Scene
5. Paste the Game Objects in the *SampleSceneAR*
6. Right click on the *SampleScene* -> select Remove Scene



# ADD ARFOUNDATION PACKAGE

7. Go to Window > Package Manager
8. In the Package Manager select “Packages: Unity Registry” then search the term AR
9. Install ARFoundation
10. Install ARKit or ARCore, depending on your target device (for this exercise we will install ARKit)

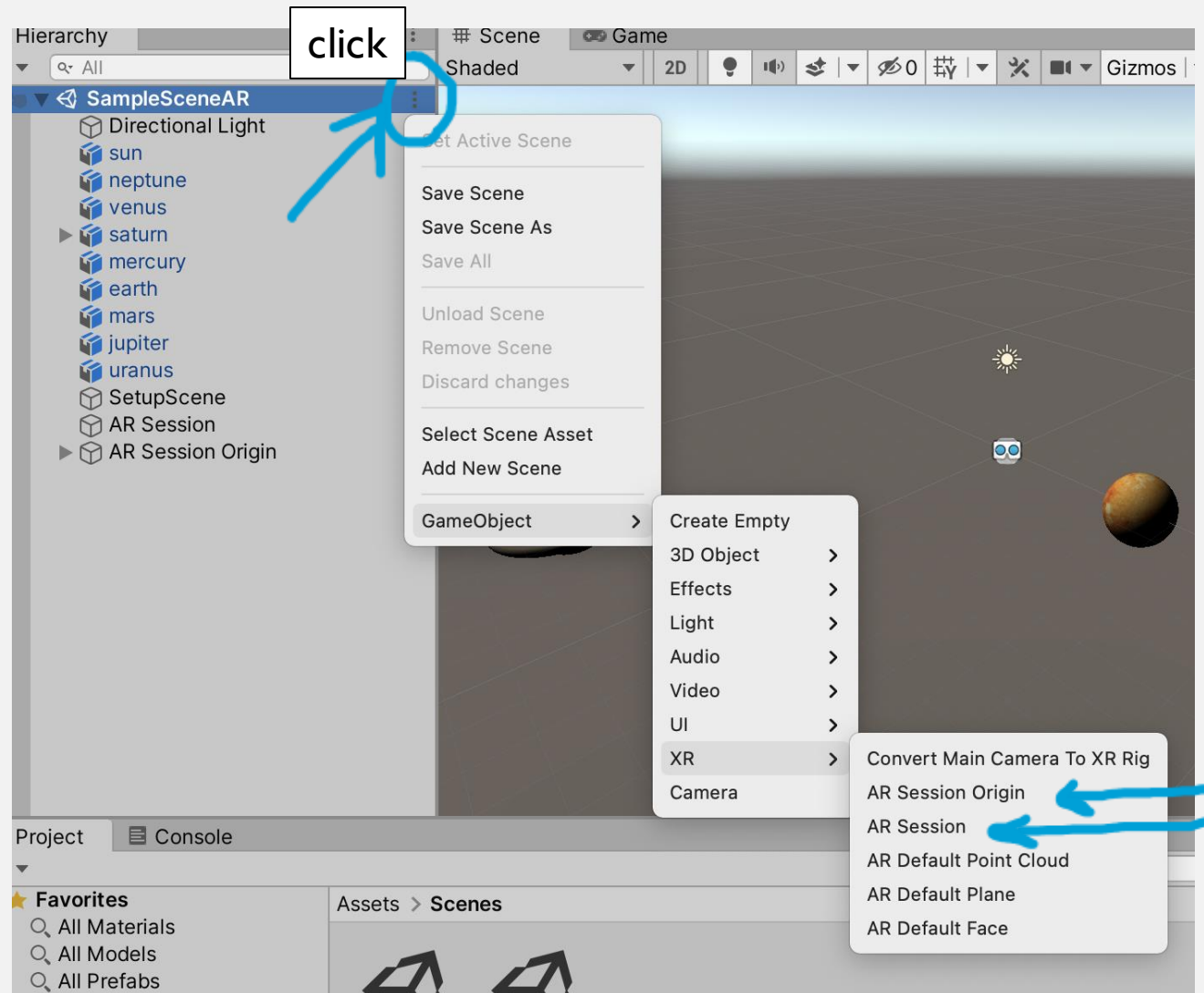


# SETUP THE *SampleSceneAR*

12. Now in the *SampleSceneAR* delete the Main Camera

13. Add a GameObject > XR > AR Session  
AR Session supports enabling and disabling XR on the target device

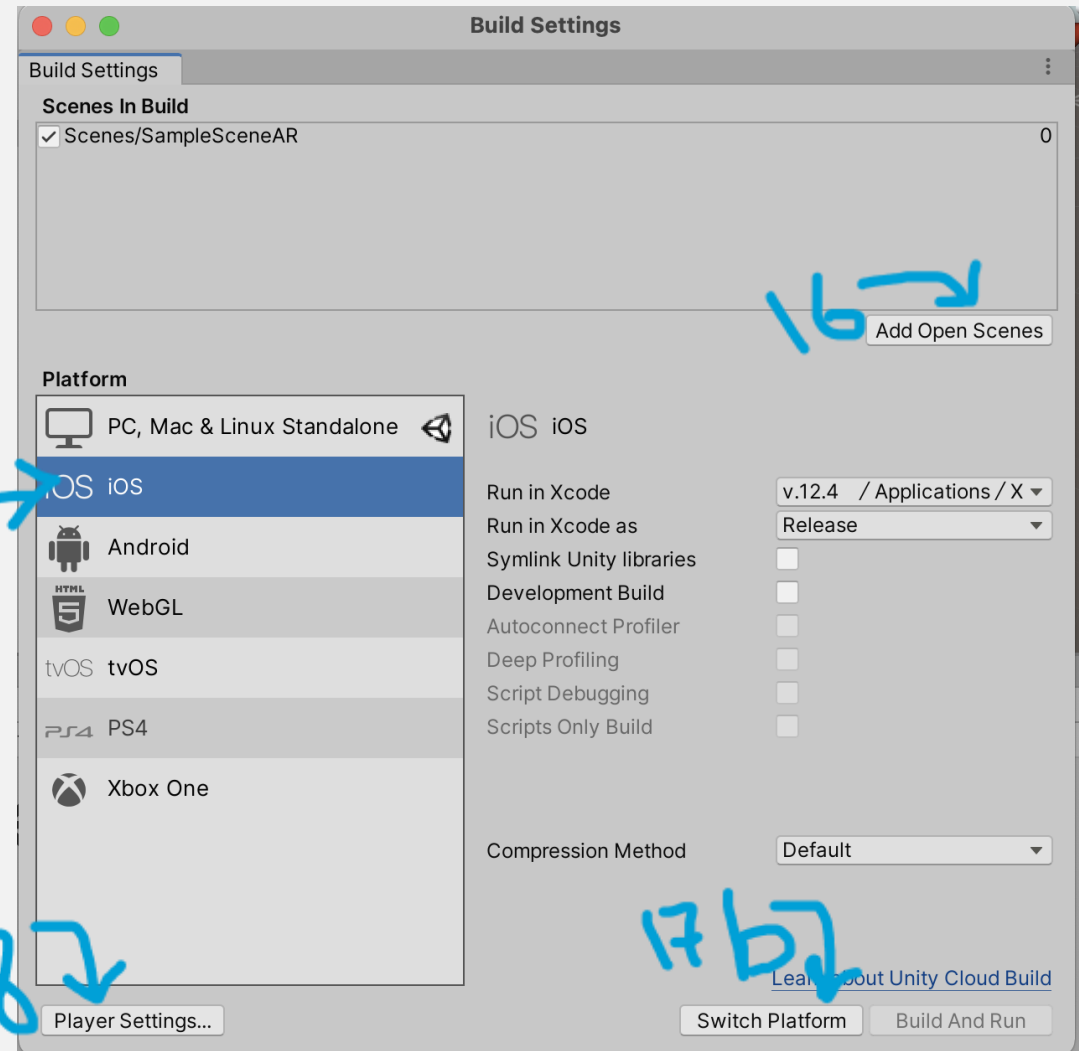
14. Add a GameObject > XR > AR Origin  
Because AR devices provide their data in "session space", which is an unscaled space relative to the beginning of the AR session, the ARSessionOrigin performs the appropriate transformation into Unity space





# BUILD SETUP

15. Go to File > Build Settings
16. Click on the button “Add Open Scenes”
17. Click on the target device iOS and then click on the button “Switch Platform,” wait for Unity to switch (it may take some minutes)
18. Now in the Build Settings click on the button “Player Settings” to open the corresponding panel



# PLAYER SETTINGS SETUP

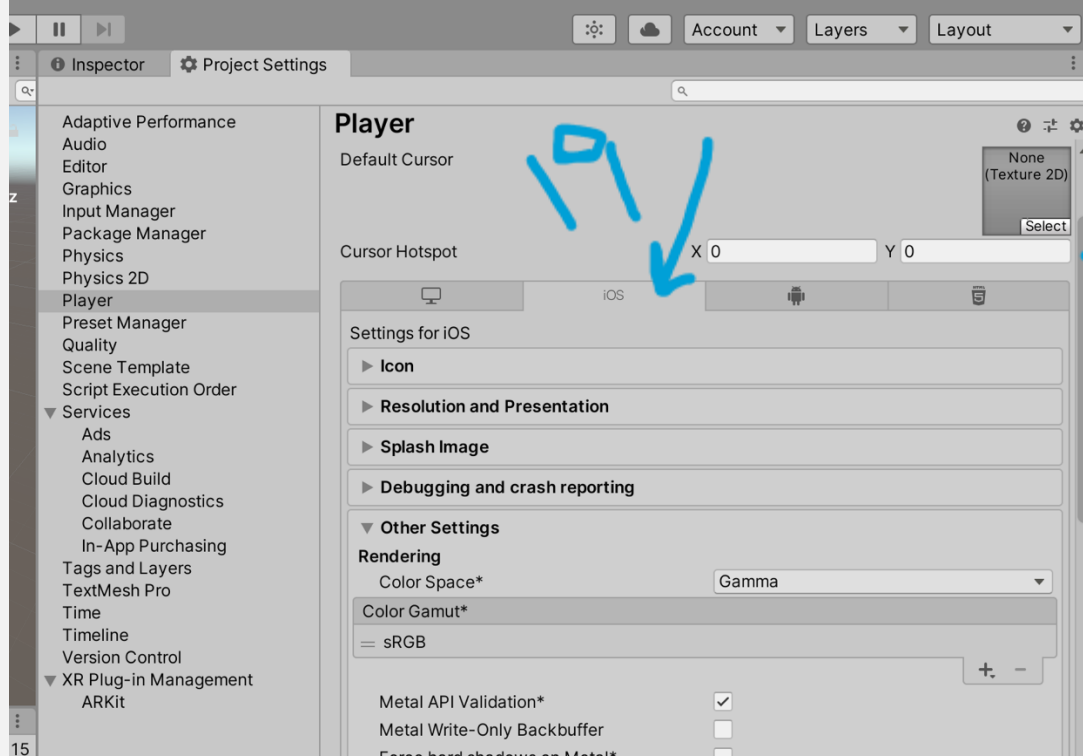
19. In the Player Settings click on iOS

20. under the voice “Other Settings” → “Configuration” check the box corresponding to “Requires ARKit support”

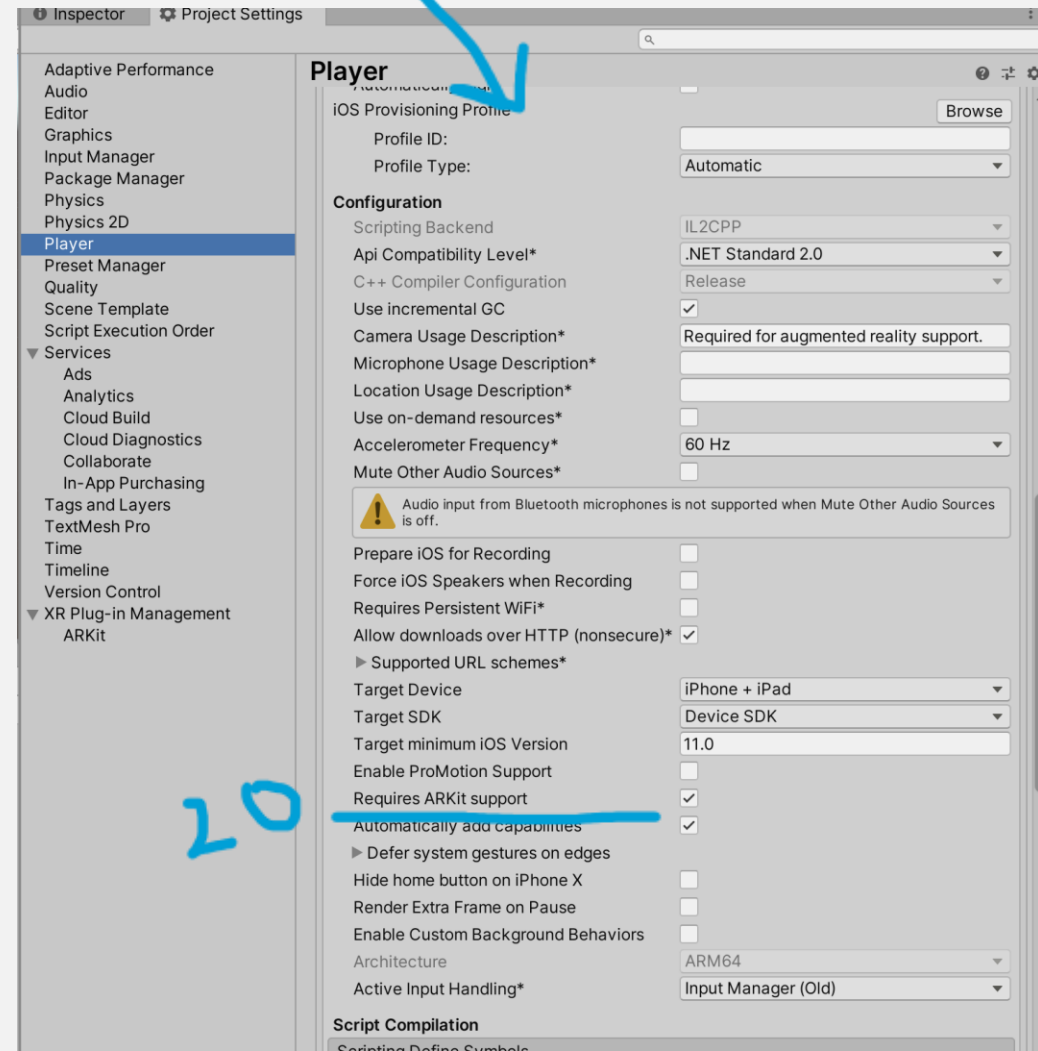
( Player Settings are inside the Project Settings window, you can also go to Player Settings by click on Edit > Project Settings then click on “Player” )



<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#using-ar-foundation>



Scrolling down with the mouse

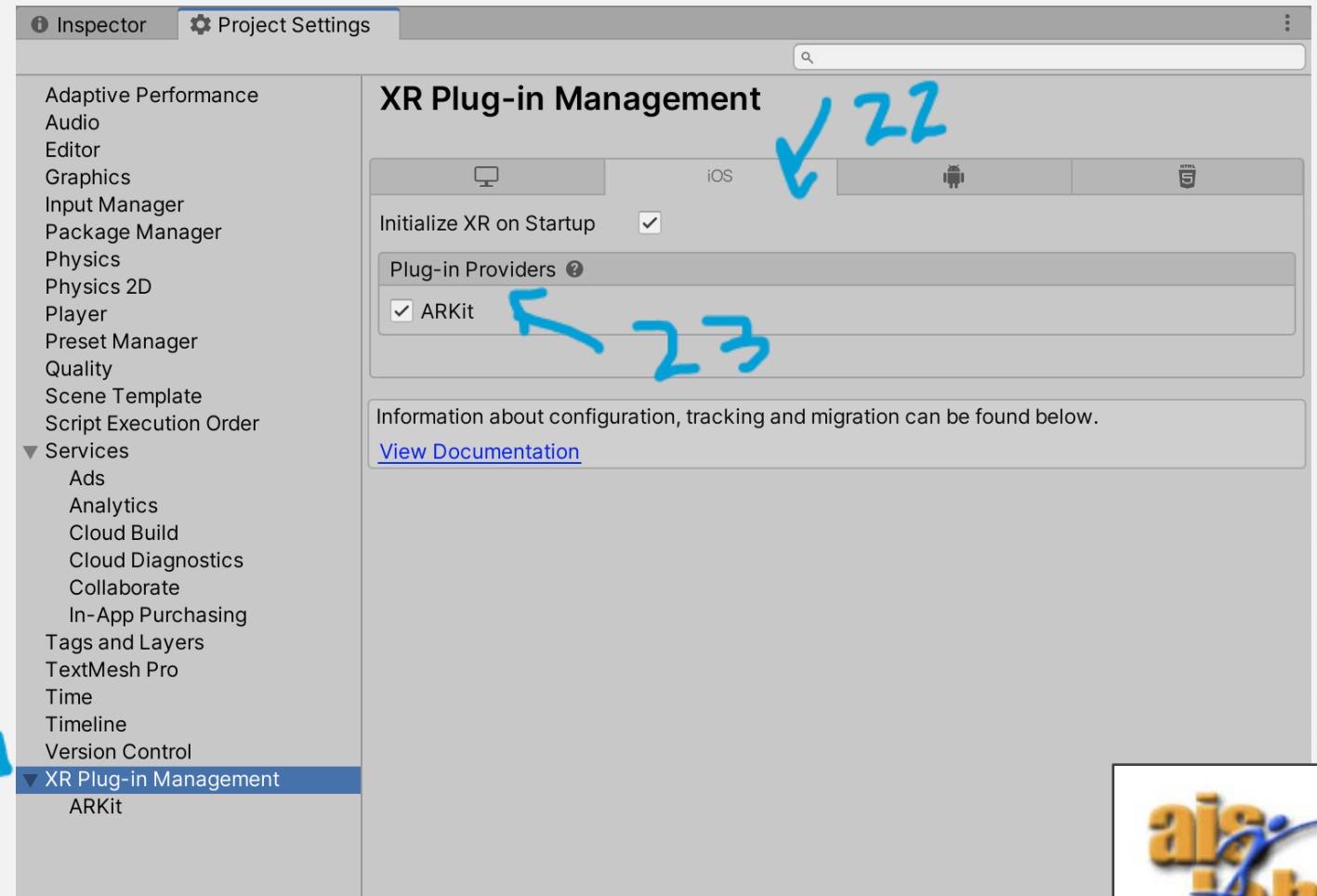


20

# XR PLUGIN MANAGEMENT SETUP

21. Now in the Project Settings click on “XR Plug-in Management”
22. Click on iOS (or Android depending on target device, in this exercise we are using iOS)
23. Enable the Provider Plugin setup, checking the box ARKit for iOS (ARCore for Android)

<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#provider-plugin-setup>

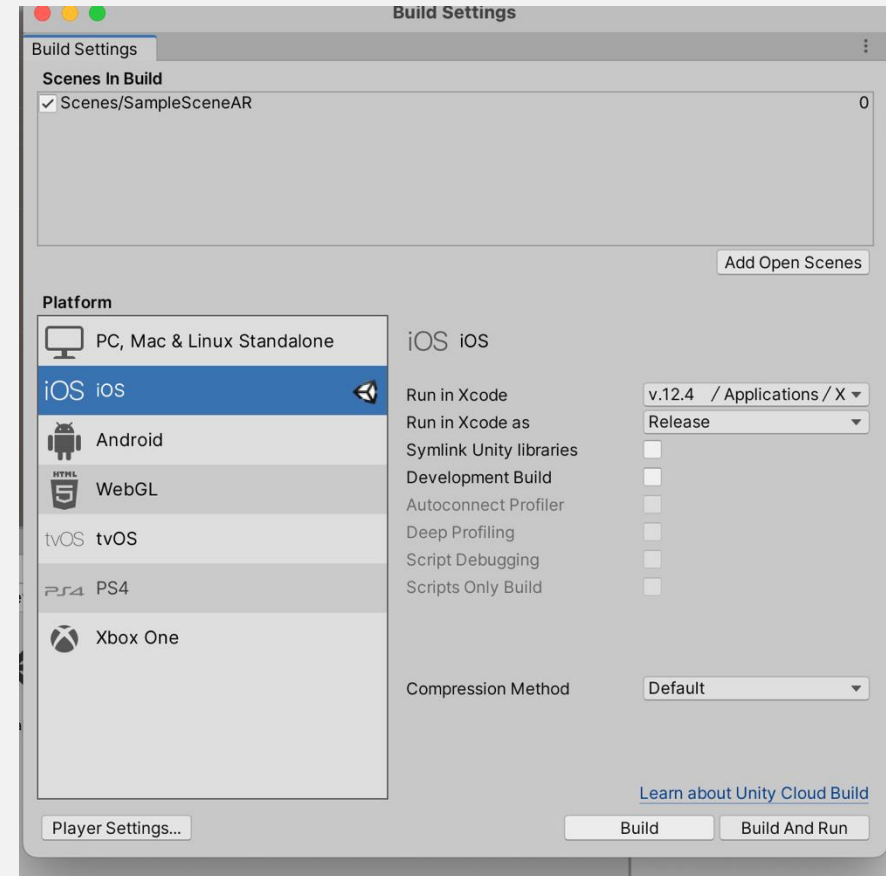


<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#using-ar-foundation>



# BUILD THE PROJECT

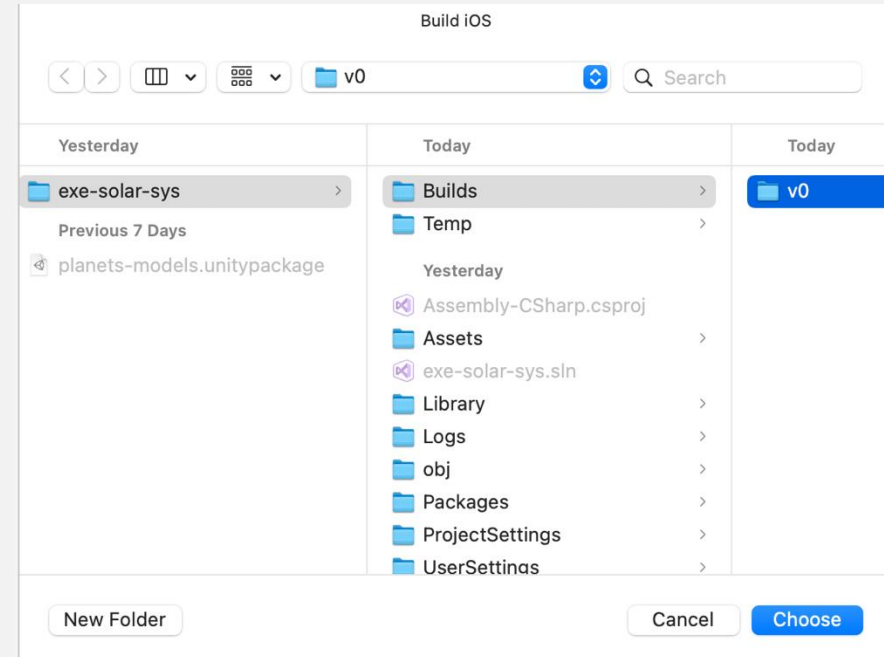
24. (Optional) Before building the project in the Player Settings under the voice “Other Settings” → “Identification” you can change the Bundle Identifier, that should be a unique identifier of your project.
25. Now you can go to Build Settings and click on Build



<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#using-ar-foundation>

# BUILD THE PROJECT

26. Now a Save Panel appears, create a folder called Builds inside the root folder called “exe-solar-sys” and inside it create another folder, we will call it “version0” :  
this folder will be the one to be opened with Xcode.

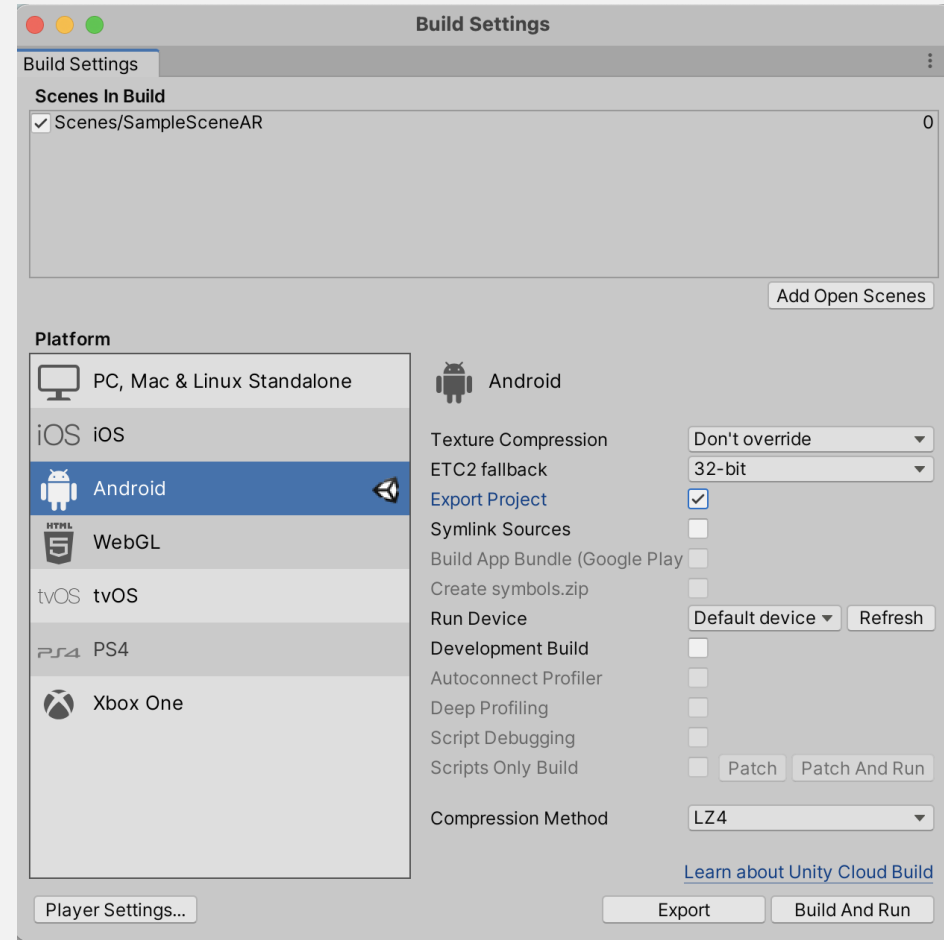


# IF YOU WANT TO BUILD WITH ANDROID

With Unity targeting an Android, you can press the button Build to directly create the .apk to be installed into your mobile device.

Otherwise, you can decide to export it as Gradle Project.

This second option is very similar to the step 26 : you need to create a Builds folder and then another folder as version0 inside it, and the version0 folder will be the one you will need to open with Android Studio.



<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html#using-ar-foundation>

# XCODE FOR IOS

To build an Xcode project for development you need a simple Apple ID (it is the free personal ID you use to login in the smartphone or on iCloud)

<https://support.apple.com/it-it/apple-id>

The free Apple ID can be used only to build TEST APP on your iOS device, it cannot be used to load on Apple Store (it needs a billing account)

To build the TEST app on the iOS target device remember that:

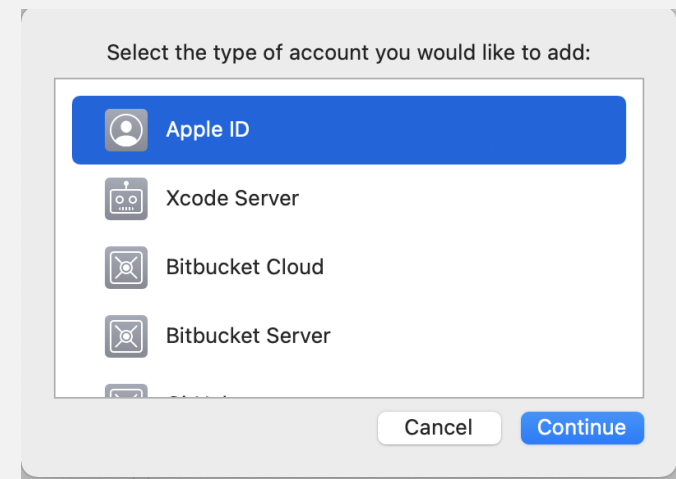
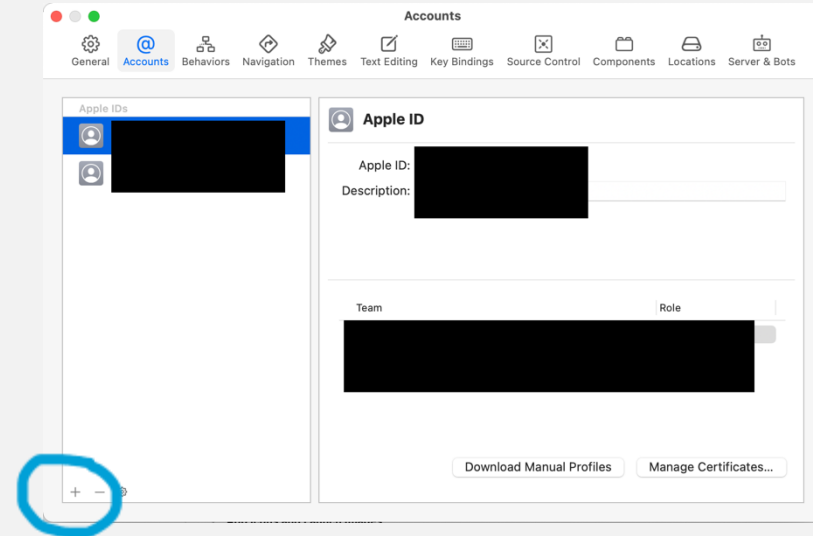
- the Apple ID on Xcode should be the same ID on the iOS device
- on the iOS device you can install no more than 3 different TEST apps





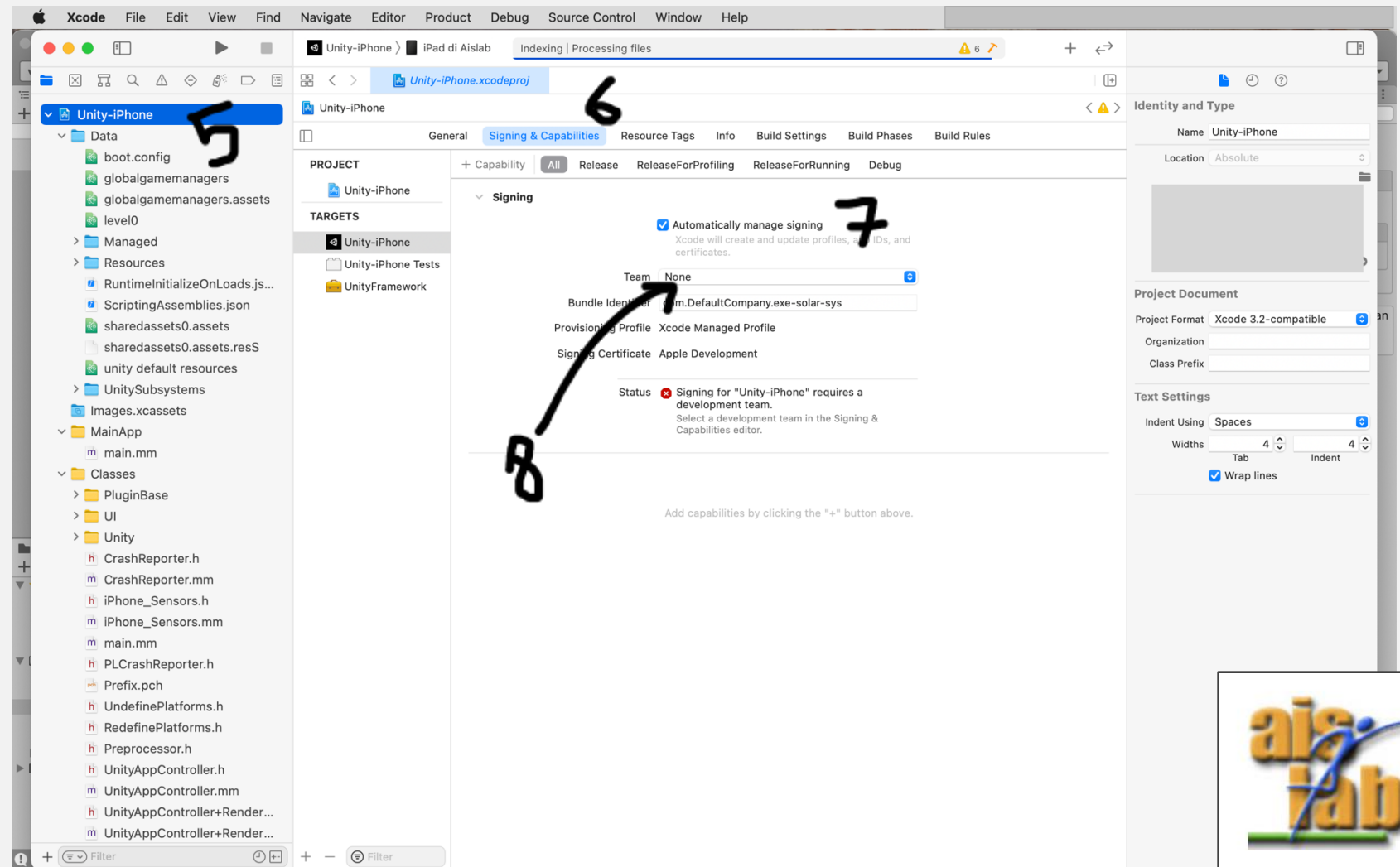
# XCODE FOR IOS

1. Once the build has done you can open the version0 folder with Xcode
2. In the menu click on Xcode > Preferences
3. Click on Accounts area and click on +
4. From the popup choose Apple ID, then click on Confirm and perform the login with your Apple ID credentials



# XCODE FOR IOS

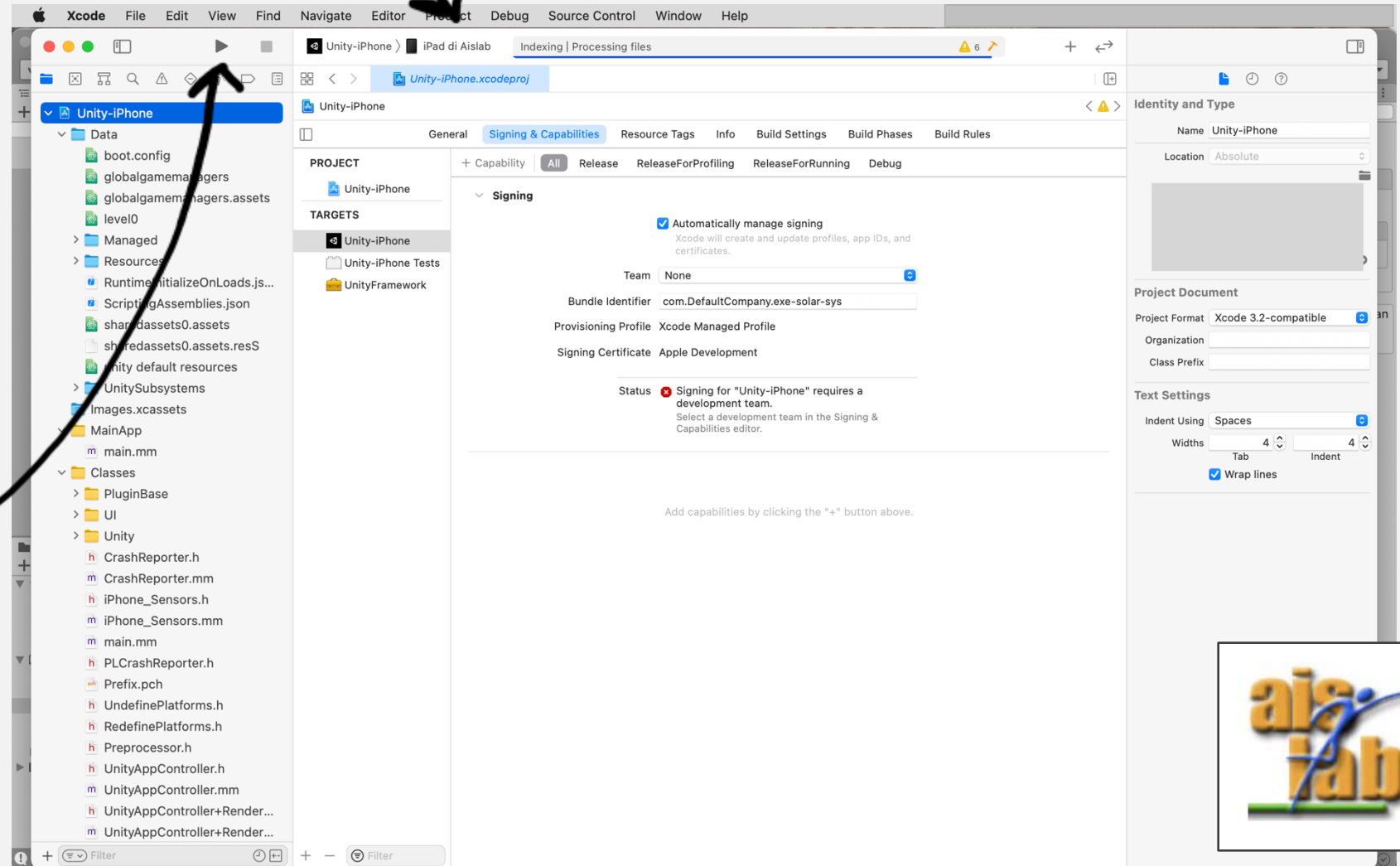
- Now click on the Root of the Project (the one usually called Unity-iPhone)
- Click on “Signing & Capabilities”
- check the box “Automatically manage signing” a popup will appear, click on the button “Enable Automatic”
- Now at the voice Team, click on “none”, and from the menu select your ID



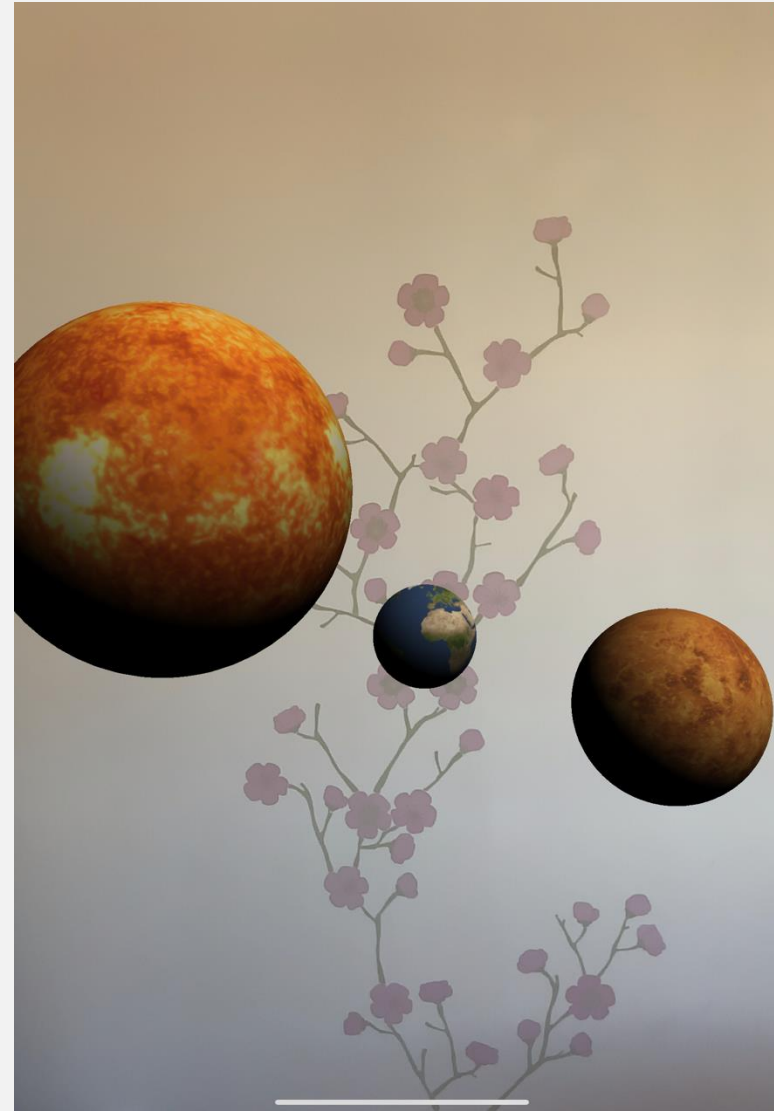
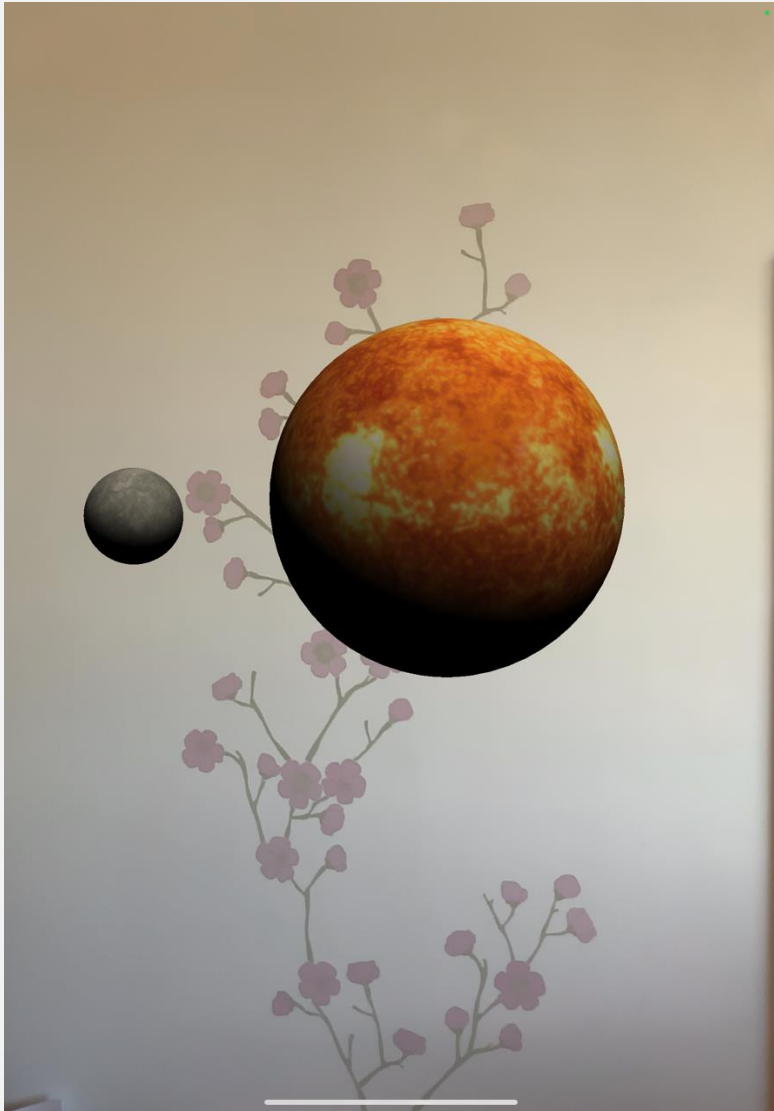
# XCODE FOR IOS

5. Now connect your device, it should automatically appear. (If it does not appear automatically you can click on “Any Device” to open the menu and see the available devices, sometimes if the device is not appearing you can just unplug it, unlock the device and then plug it again)

6. Now you can Run it



# SCREENSHOT OF THE FINAL RESULT 😊



# TROUBLESHOOTING

- [iOS / Android] If after the build, you run it on your device and the screen is black: you should enable the Provider Plugin setup, in the “Project Settings” click on “XR Plug-in Management” then click on iOS or on Android and respectively check the box ARKit or ARCore ( step 11)
- [Android] If after the build you get an error on ARCore as “BuildFailedException:ARCore Required apps require a minimum SDK version of 24.” or “Failed getting available Android API levels. Could not find Android SDK Tools.”

You can solve it: Project Settings > Player Settings > Android, under the voice “Other Settings” → “Identification” change the “Minimum API Level” to Nougat level 24

- [Android] “BuildFailedException:You have enabled the Vulkan graphics API, which is not supported by ARCore.”
  - Under the voice “Other Settings” → “Rendering” change the Graphics API and remove Vulkan.
  - If you can not find Vulkan in Rendering: In BuildSettings -> Project settings -> Player -> Other settings: disable the **Auto Graphics API checkbox** and remove Vulkan from the Graphics APIs list  
<https://forum.unity.com/threads/you-have-enabled-the-vulkan-graphics-api-which-is-not-supported-by-arcore.896741/>

