

Easy AR

Easy AR Documentation

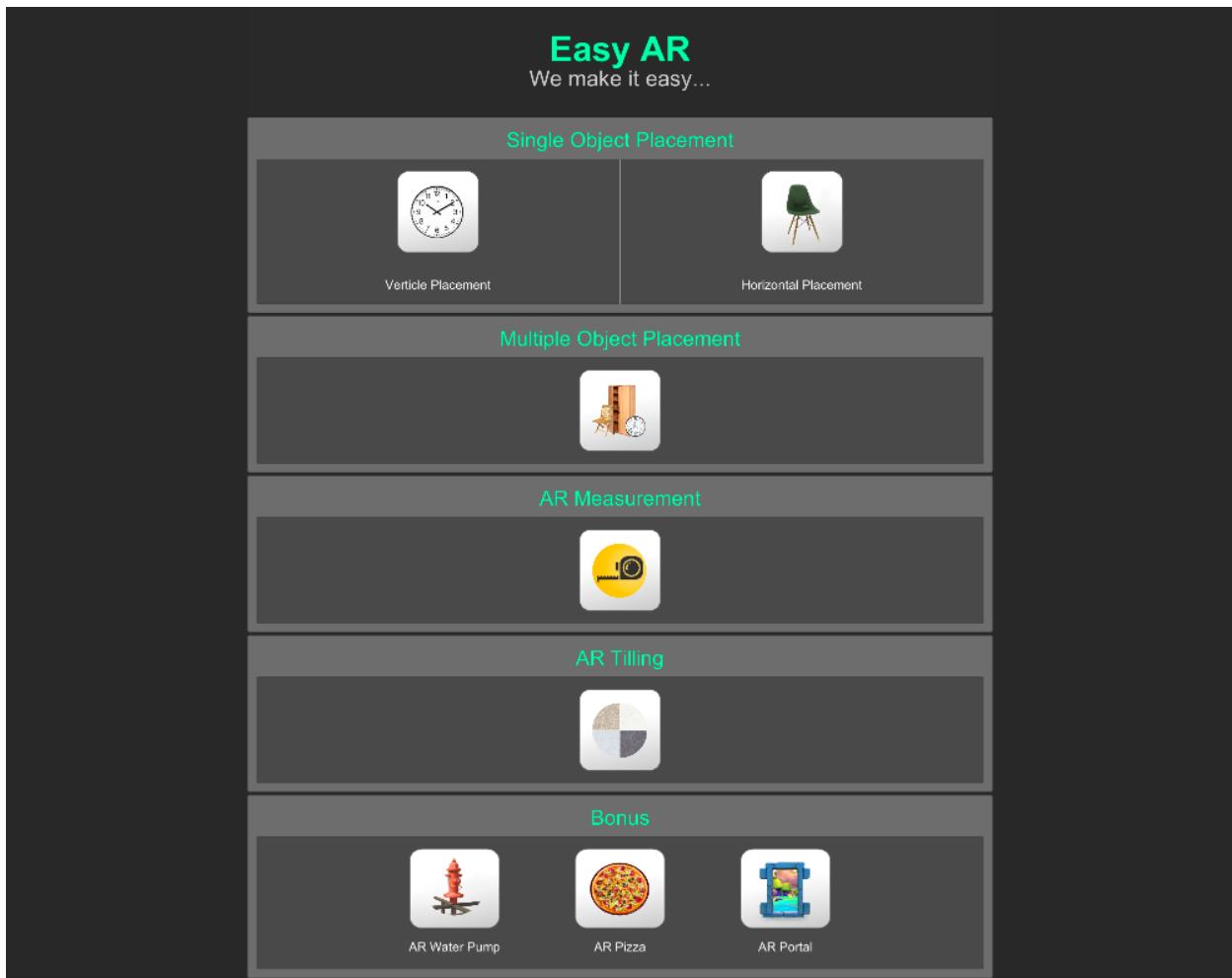
1. Introduction
2. Features in Easy AR
 - 2.1. Single placement
 - 2.1.1. Vertical placement
 - 2.1.2. Horizontal placement
 - 2.2. Multiple placement
 - 2.3. AR Measurement
 - 2.4. AR Tilling
 - 2.5. Bonus
 - 2.5.1. AR Water pump
 - 2.5.2. AR Pizza
 - 2.5.3. AR Portal
3. Import Easy AR package in to Unity
 - 3.1. Unity version
 - 3.2. AR Foundation version
 - 3.3. AR Core XR plugin version
 - 3.4. AR Kit XR plugin version
4. Setting up the default demo scenes
5. Build the Application
6. How to create a new object prefab
7. Setting up a custom scenes
 - 7.1. How to set up the Single Object Placement
 - 7.2. How to set up the Multiple Object Placement
 - 7.3. How to set up the AR Measurement
 - 7.4. How to set up the Tilling
8. Shadow map generation

1. Introduction

Easy AR is an Unity package which can be used to easily set up your own Augmented Reality application without any coding skills. We have improved features of Easy AR with AR Foundation 4+ Versions & brings you 8 new utility scenes which are AR Single Vertical & Horizontal Placement, Multiple Object Placement, AR Measurement, AR Tilling, AR Water Pump, AR Pizza & AR Portal.

2. Features in Easy AR

- 1) Automatic Object Placement
- 2) Configurable Gesture Controls
- 3) Automatic Lights & Reflections
- 4) Baked Contact Shadows
- 5) No Coding Skills Needed
- 6) Compatible with iOS and Android
- 7) AR Measurement Utilities
- 8) AR Tilling Utilities
- 9) Bonus Configurable Utility Scenes



Easy AR introduces a new UI for the main menu due to enhancement of feature demo. Each & every feature demos explained below in detail.

2.1. Single Object Placement

As you can see, the first segment of Easy AR, single object placement, is divided into two. Those are vertical object placement (clock) & Horizontal object placement (chair).

2.1.1. Vertical placement



The objects that are placed in vertical space such as clock, wall paintings can be experienced from this feature. For example, here it uses a clock to place on the wall.

2.1.2. Horizontal Placement



The objects that are placed in horizontal space can be experienced using horizontal placement. For example, the green chair is added here.

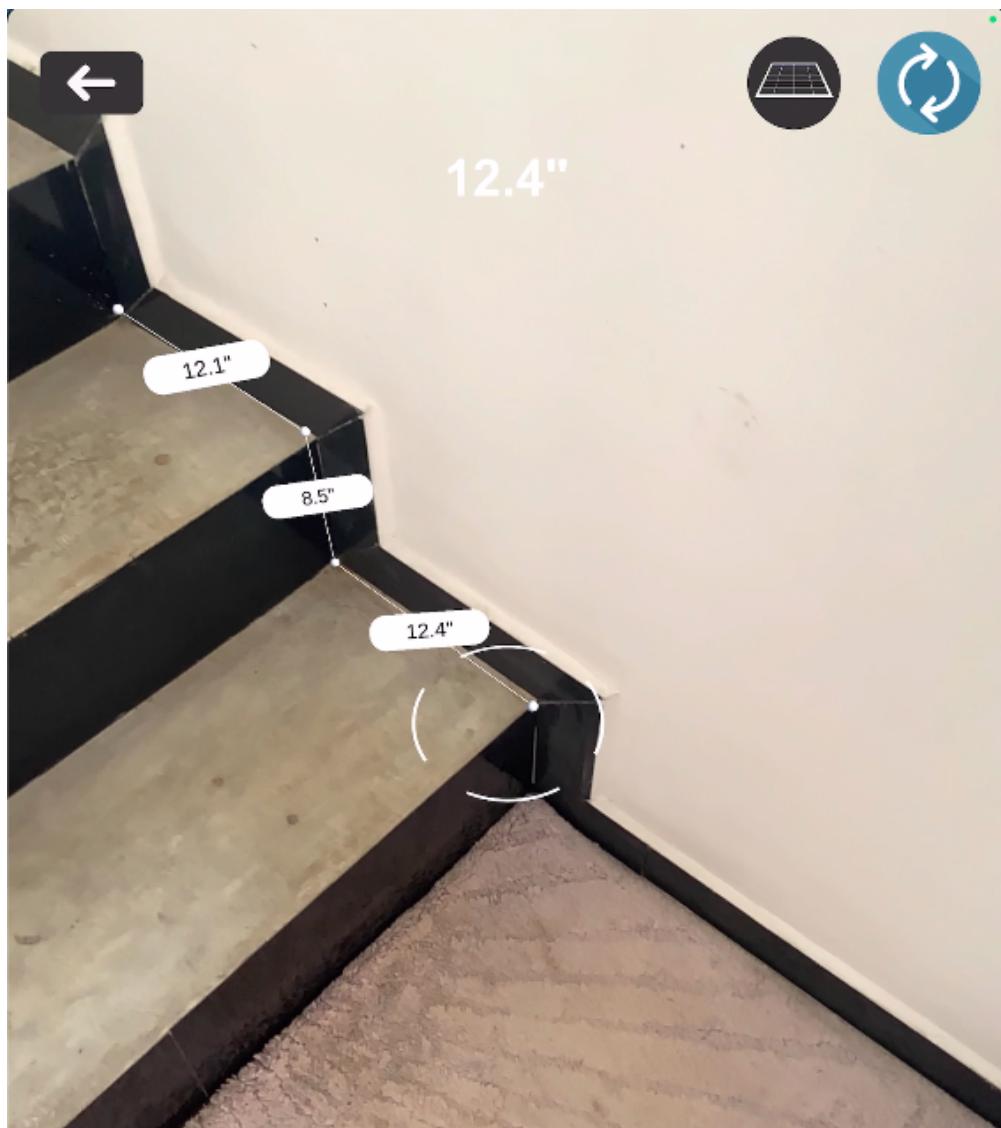
2.2. Multiple object placement



Multiple object placement uses both horizontal placement & vertical placement at the same time. From this demonstration users can place virtual objects on desired spots in a room or empty space that you have. In multiple placement demo users can place the same item several items & sometime users can place multiple objects. For the demo we have given you a golden elephant statue, wood chair, sofa, wall painting & clock. In each item set their placement as vertical placement item or horizontal placement item. According to them, the user can place them in identified planes. It

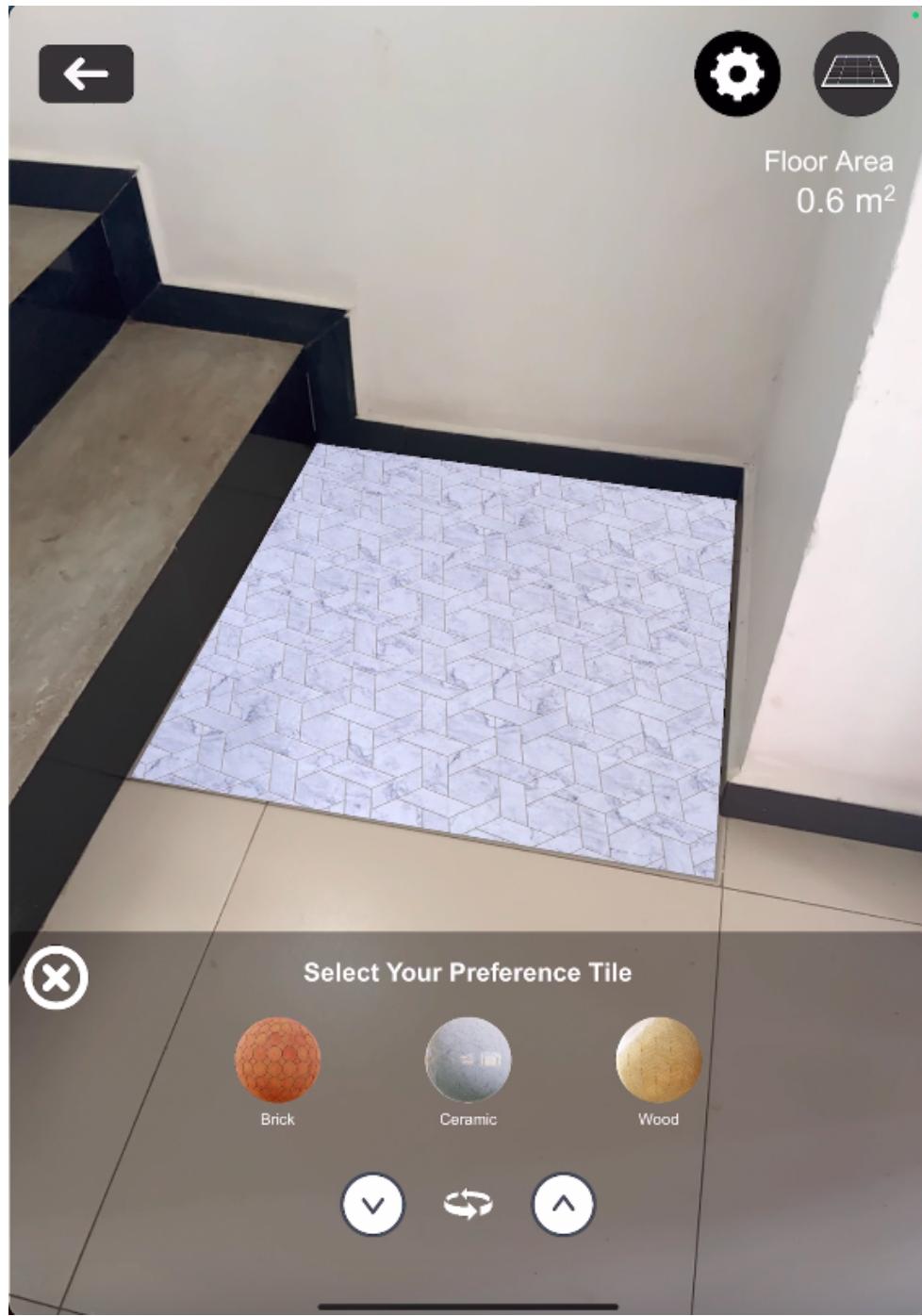
doesn't allow you to place a vertical placement object in horizontal planes or a horizontal placement object in vertical planes.

2.3. AR Measurement



AR measurement allows users to measure length along horizontal or vertical space easily with use of AR technology. In each measurement it shows a card with value included line length & card rotate where mobile devices face.

2.4. AR Tiling



AR tiling is quite new as a use case & it's very easy to demonstrate to the user to choose the desired tile that they need through the AR. Users are able to mark the floor area by adding points & it generates mesh & applies tiles to

mesh. Users can customize tiles tiling & offset value along with any direction. This can be used for marketing purposes in the field of interior designing, architecture & etc

2.5. Bonus Demos

At last Easy AR presents new bonus scenes with a newer version. The purpose of presenting this demo, we hope these demos can be used in many fields in AR such as construction, marketing, entertainment & etc.

2.5.1. AR water pump



AR water pump showcases how users can see underground pipe systems connected to water pumps & adjust visibility of pipe systems. This demo is more appropriate in the construction field of AR.

2.5.2. AR pizza



AR Pizza is a customizable product which customers can add toppings as much as they need. After adding the toppings, they can process the pizza. This will help pizza backers to identify the percentage of toppings preferred by customers before baking. For AR marketing purposes this demo is more appropriate.

2.5.3. AR Portal



AR portal shows another fantasy world after the user passes through the portal. When a user stands inside the portal he/she can see that world as well as the real world through the portal entrance. This demo is suitable for the field of entertainment in Augmented Reality.

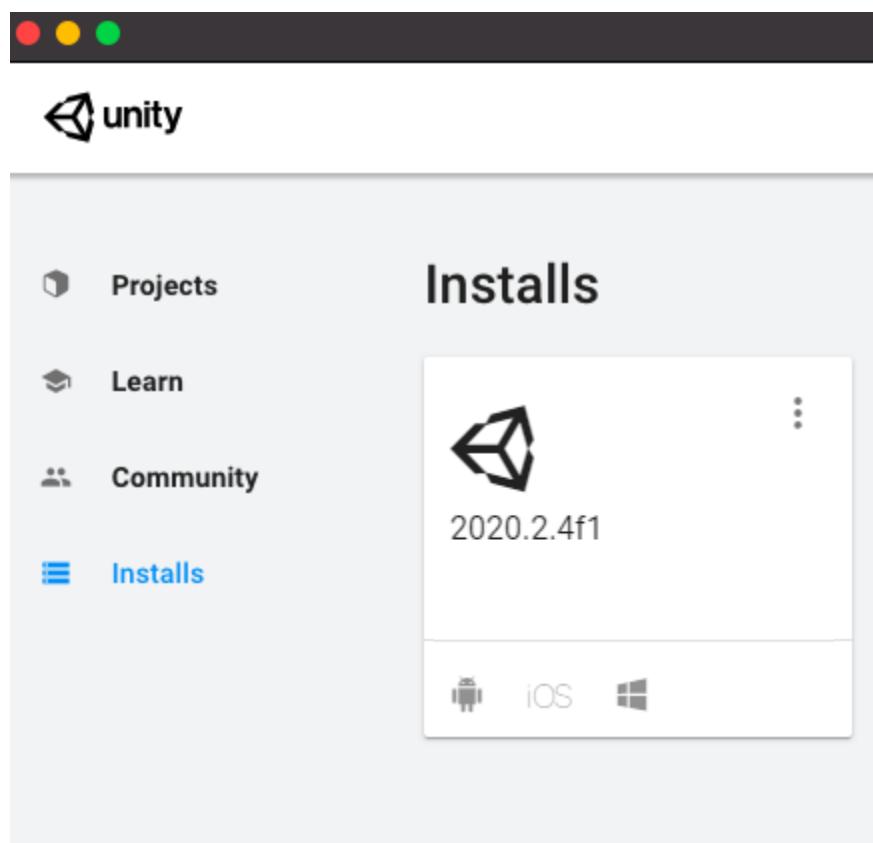
3. Import Easy AR package into Unity

We highly recommend using [Unity \(LTS\)](#) verified versions. Initially we built the package using the following Unity version and used the [AR Foundation package](#), [ARCore XR Plugin](#) and [ARKit XR Plugin](#).

3.1. Unity Version

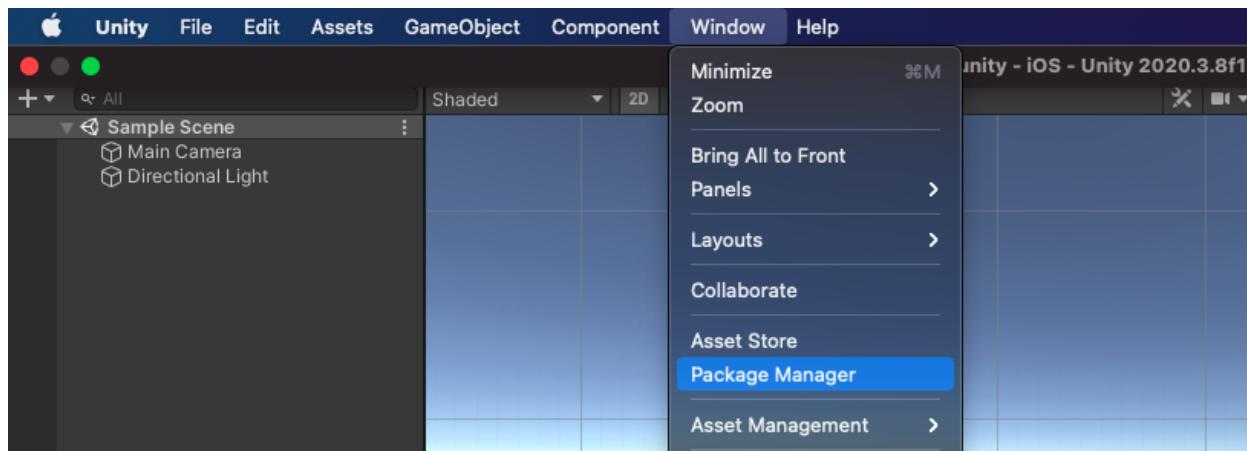
- [Unity 2020.2.4](#) or higher LTS versions

In the Unity hub, under the Installs tab you can add this verified version.



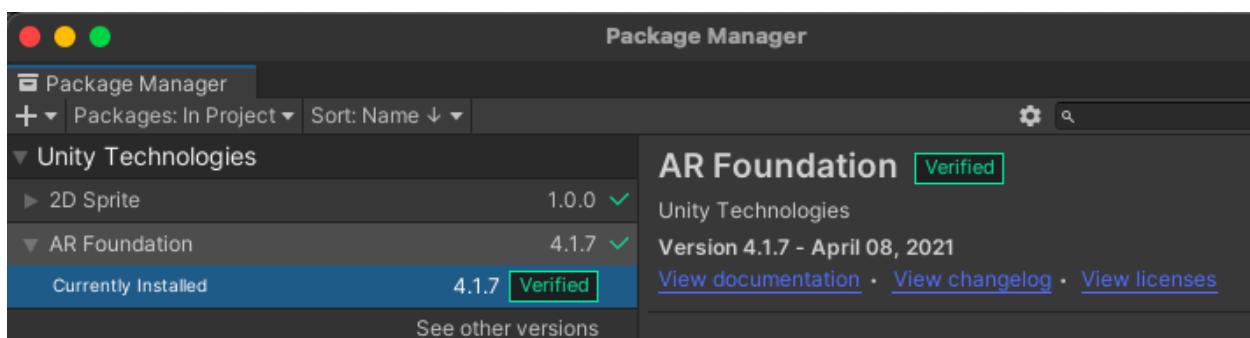
After creating a project using the above mentioned unity version and follow the below steps to import the packages.

Go to the **Window** and open the **Package Manager**. Install the following packages into the project.



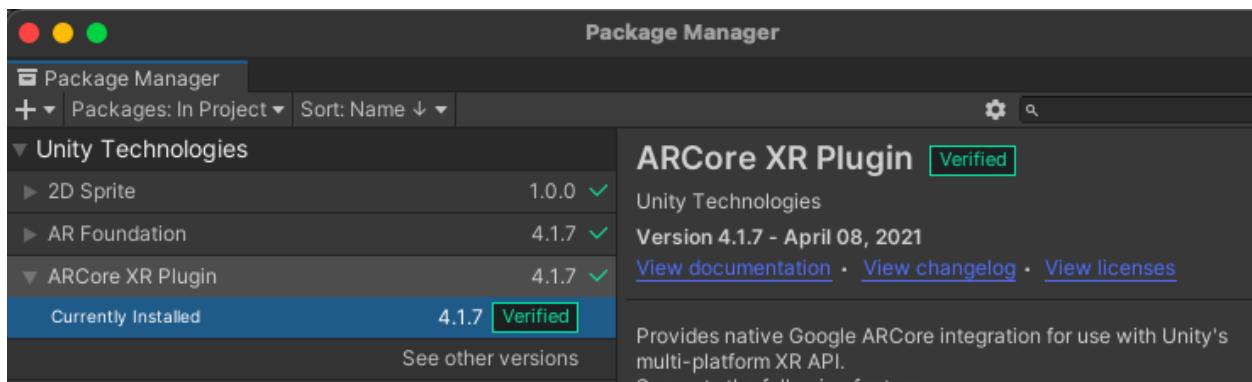
3.2. AR Foundation version

- AR Foundation package 4.1.7



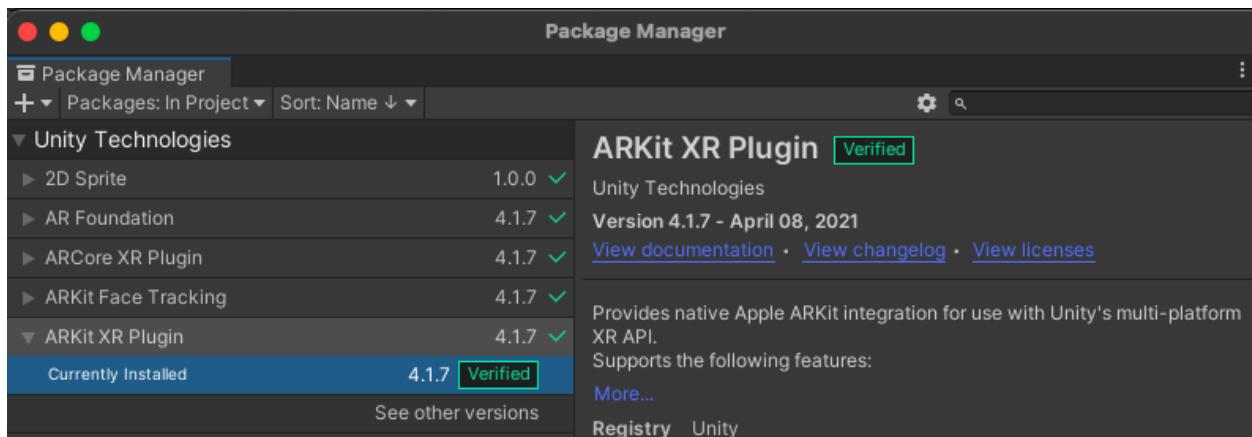
3.3. ARCore XR Plugin version

- ARCore XR Plugin 4.1.7



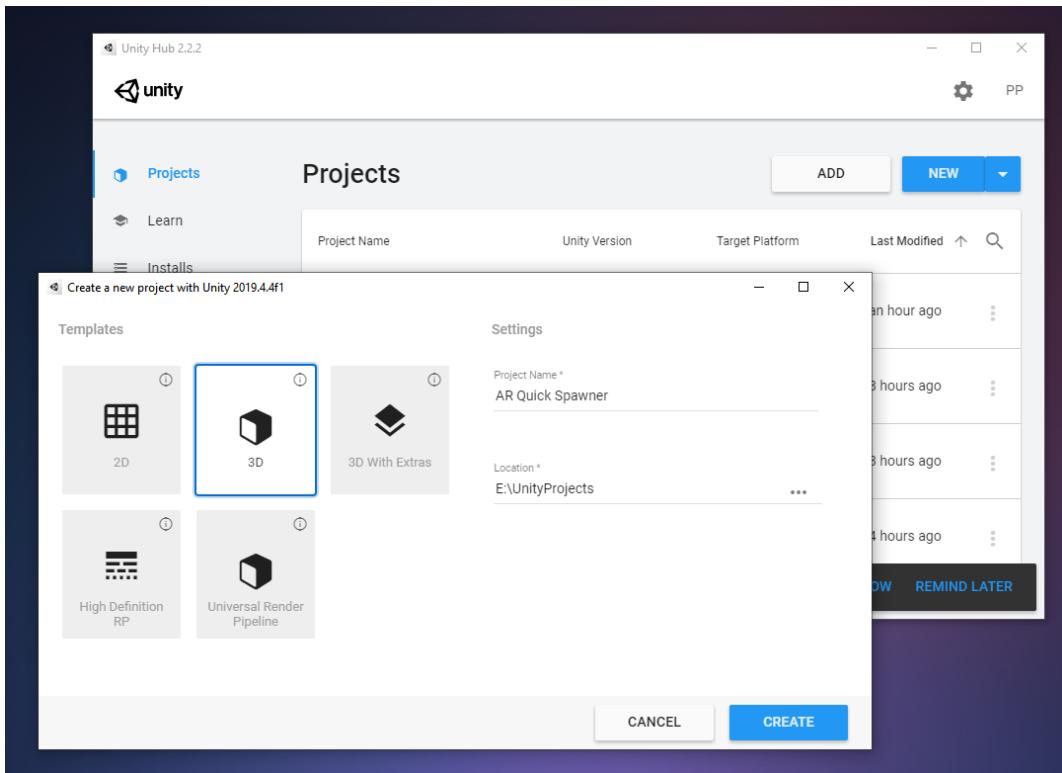
3.4. ARKit XR Plugin version

- ARKit XR Plugin 4.1.7



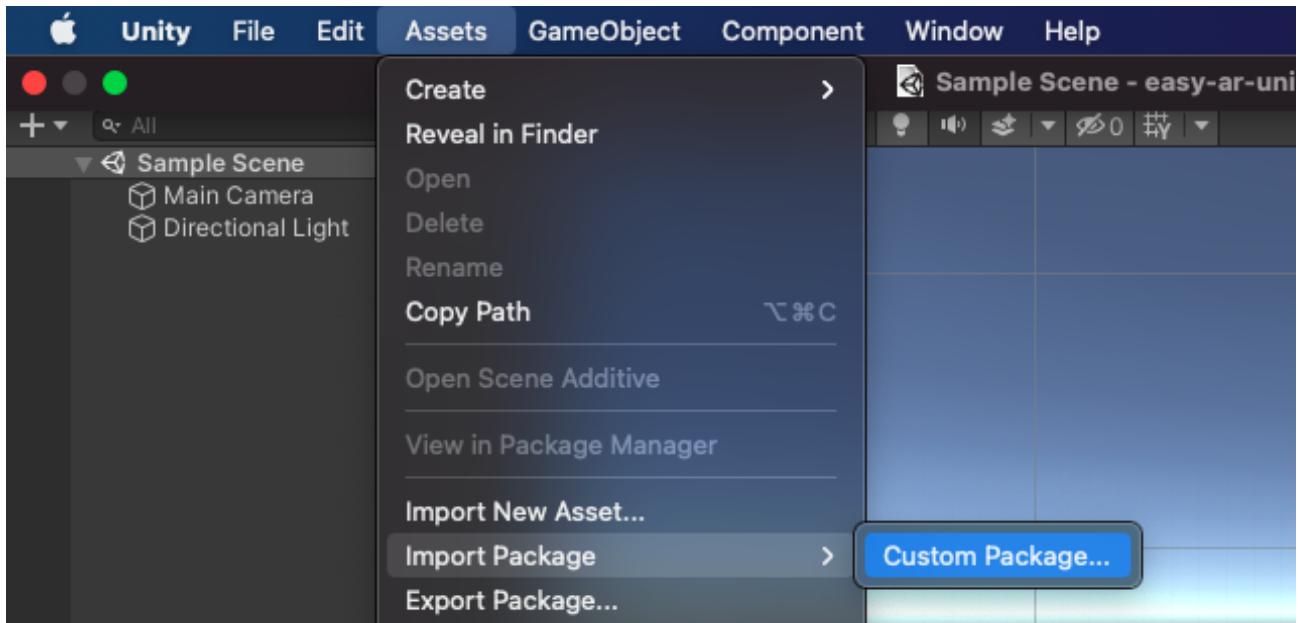
4. Setting up the default demo scenes

4.1.Create a **Unity** mentioned version of an empty project from **Unity Hub** and change **Build Platform** to either iOS or Android as desired.



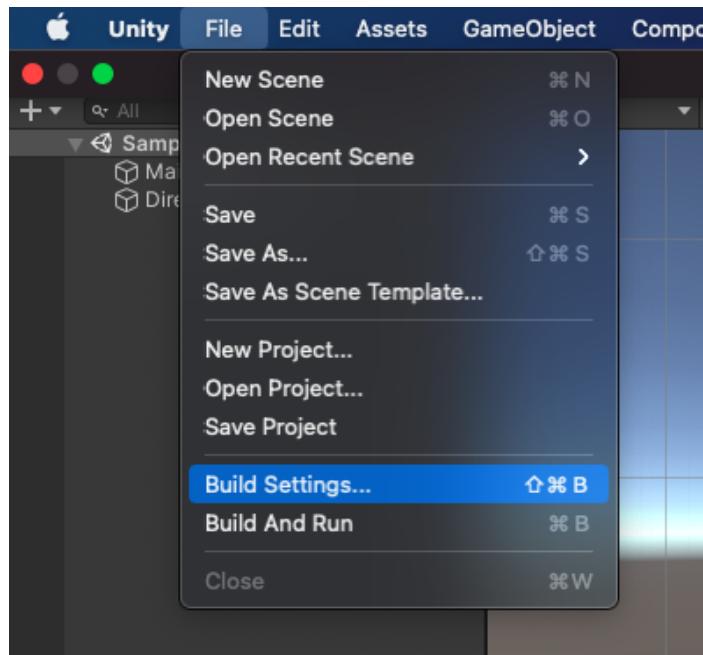
- ❖ Here you have to import the **AR Foundation**, **ARCore XR Plugin** and **ARKit XR Plugin** packages. Please follow [Section 3](#) steps if you haven't imported the packages yet.

4.2.Import the **Easy AR** package into the project.



5. Build the application

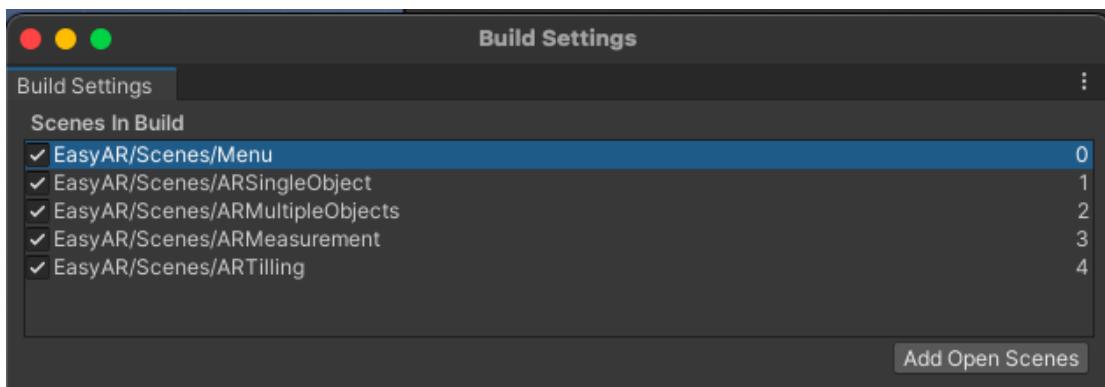
5.1.Go to File->Build settings



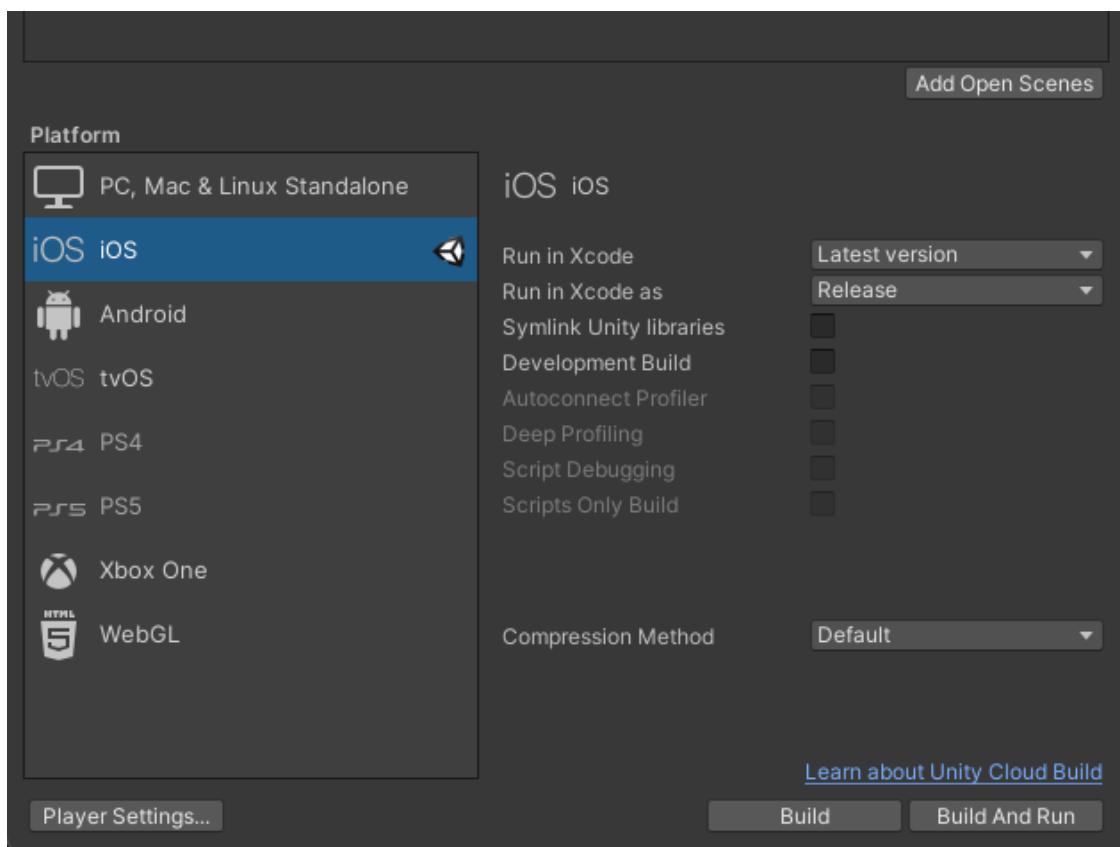
5.2. Go to **EasyAR->Scenes** folder and drag and drop the five scenes to build settings.

Build index order should be like below

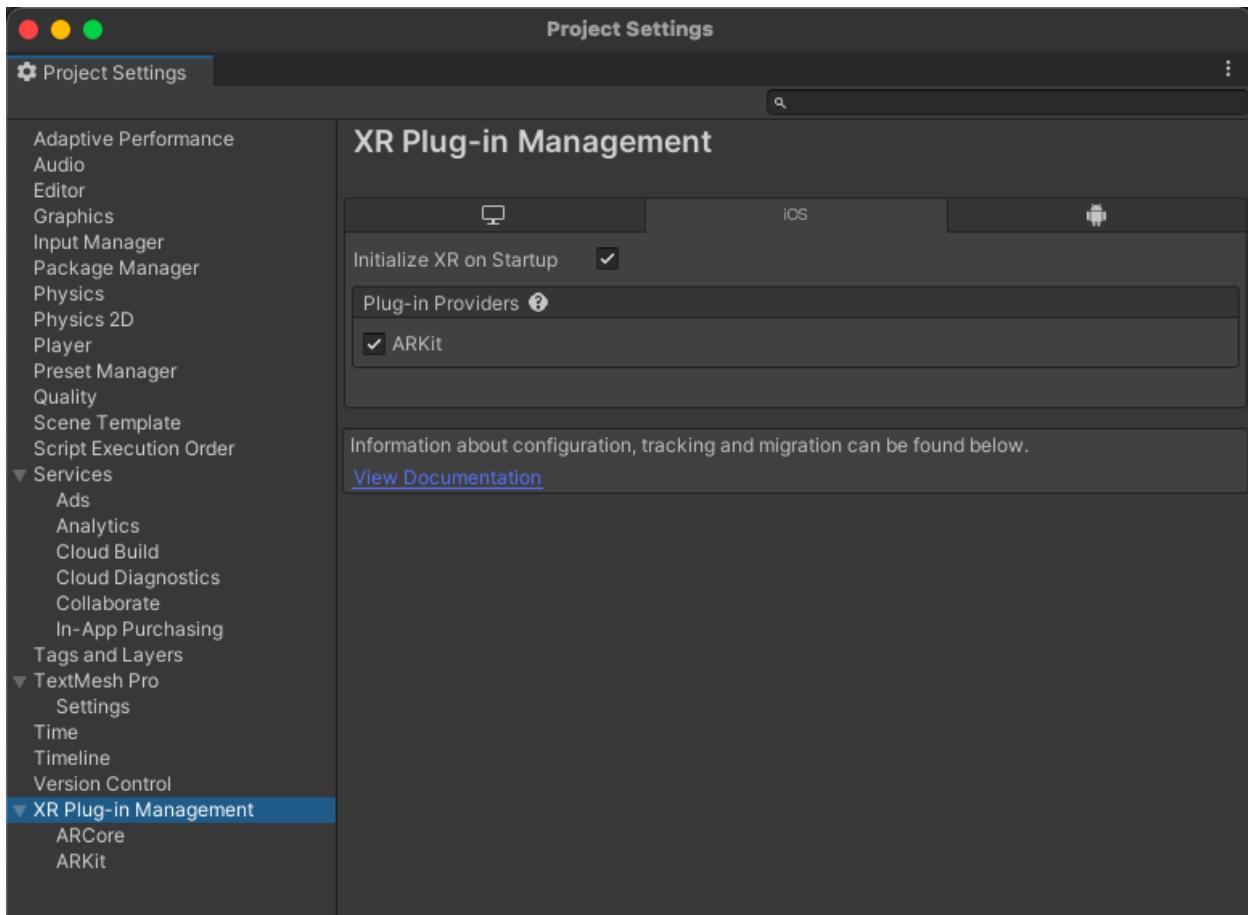
- 0** - Easy AR/Scenes/Menu
- 1** - Easy AR/Scenes/ARSingleObject
- 2** - Easy AR/Scenes/ARMultipleObjects
- 3** - Easy AR/Scenes/ARMeasurement
- 4** - Easy AR/Scenes/ARTilling



5.3. Switch target Platform you desire to build either android or IOS.



5.4. If your desired platform is iOS, switch the target platform to iOS. Go to Player setting -> XR Plug-in Management & go to IOS tab.Under plug in providers enable ARKit.



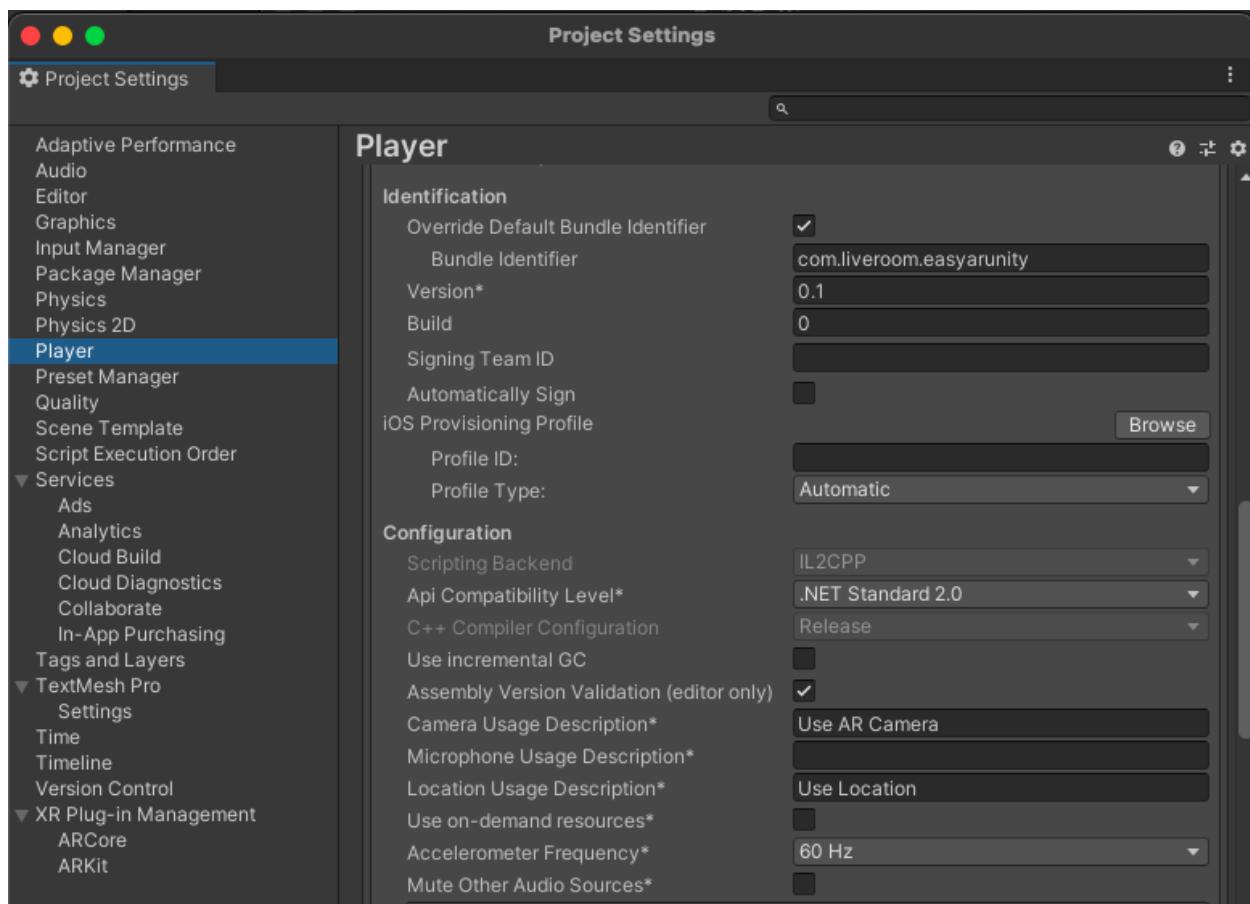
5.5. Then Go to Player setting -> Player -> Other settings in the IOS tab.

Check under Identification below context are filled

bundle identifier -> com.companyName.productName

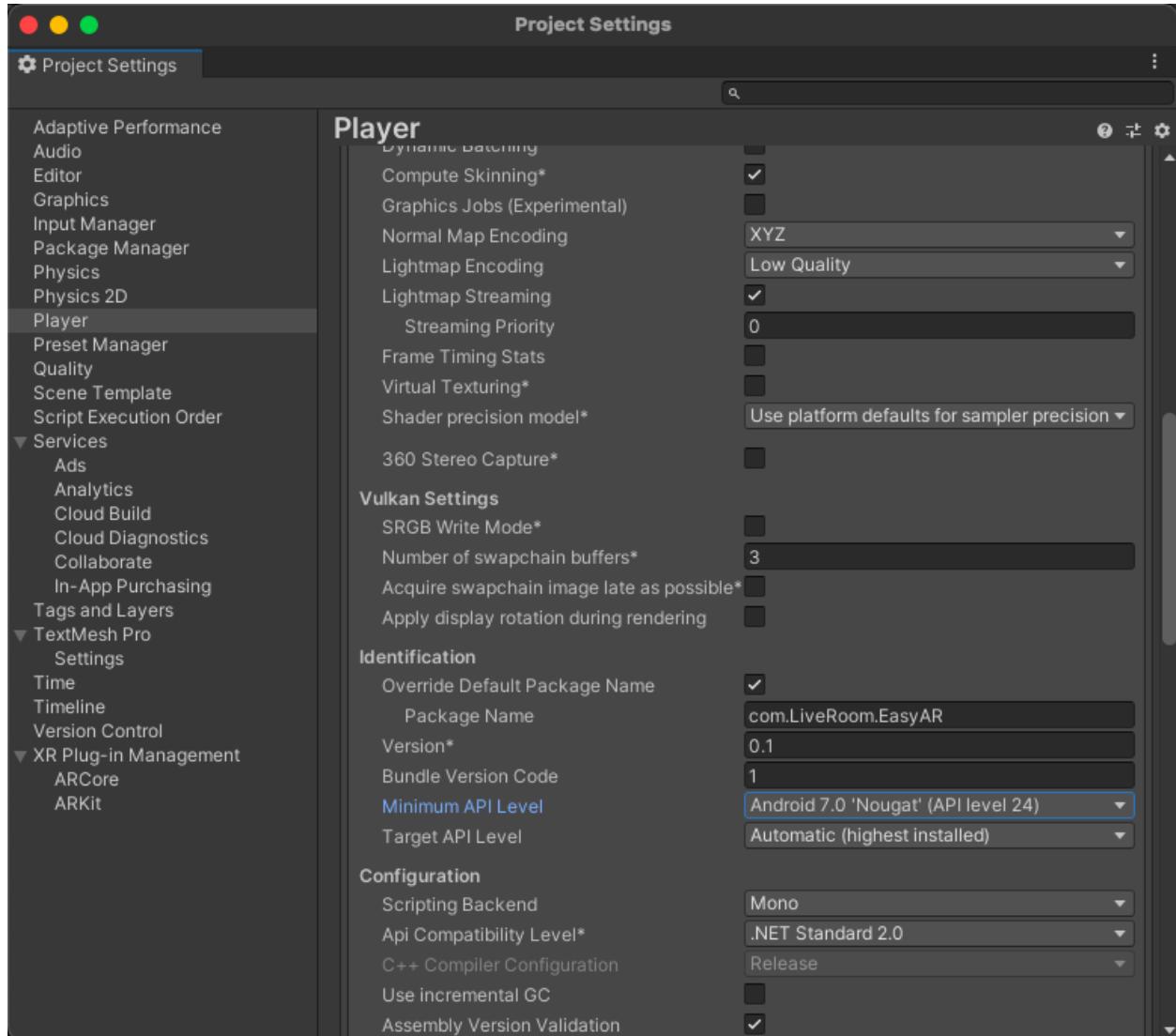
Camera usage description -> Use AR Camera

Location Usage description -> Use Location

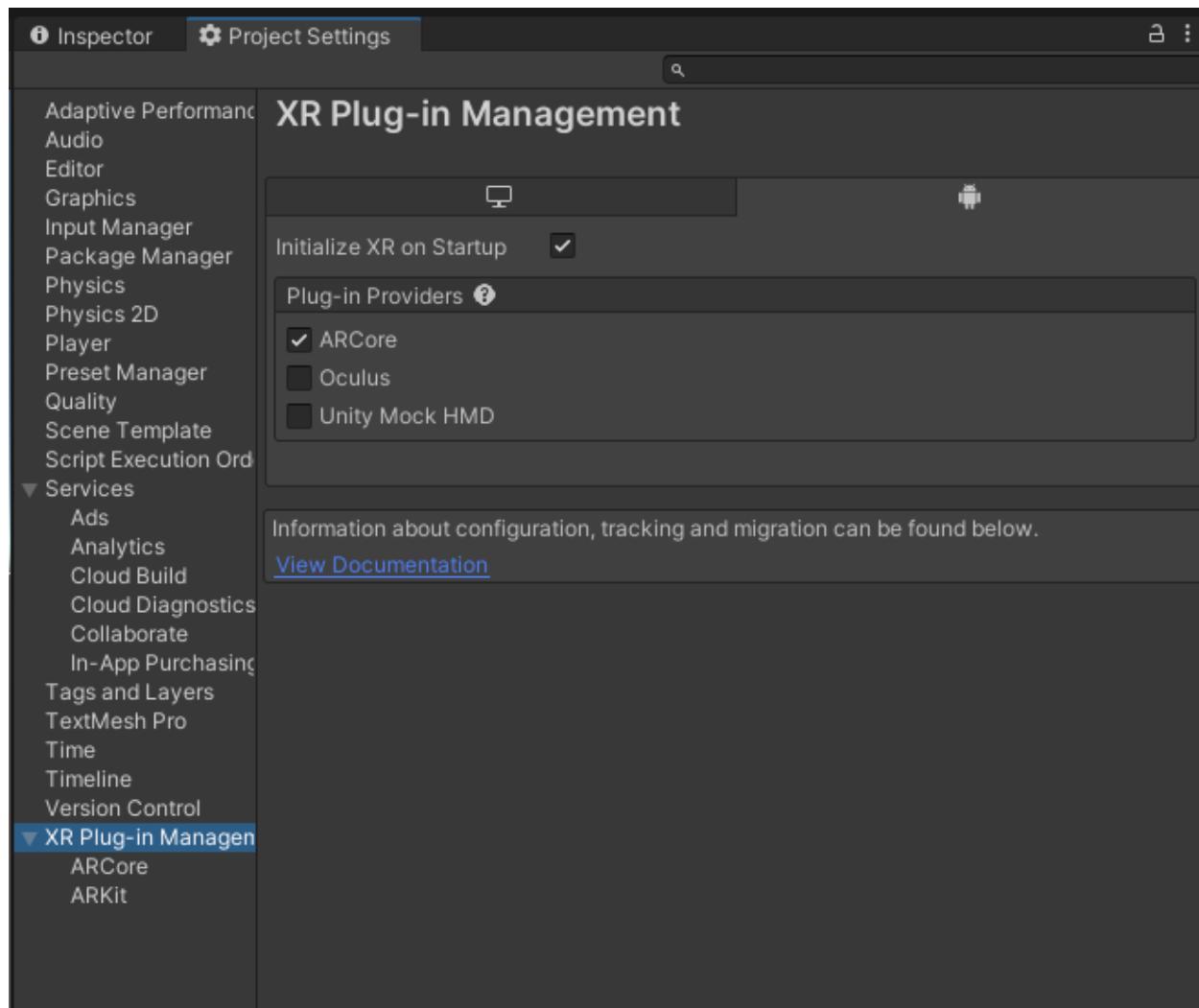


5.6. If your desired platform is **android**,

- Go to **Edit > ProjectSettings > Player > Other Settings** then set the android **Minimum API** level to **API level 24**
- Then **remove** Vulkan Graphic API



→ Go to Player Settings -> XR Plug-in Management & go to Android tab.Under plug in providers enable ARCore.

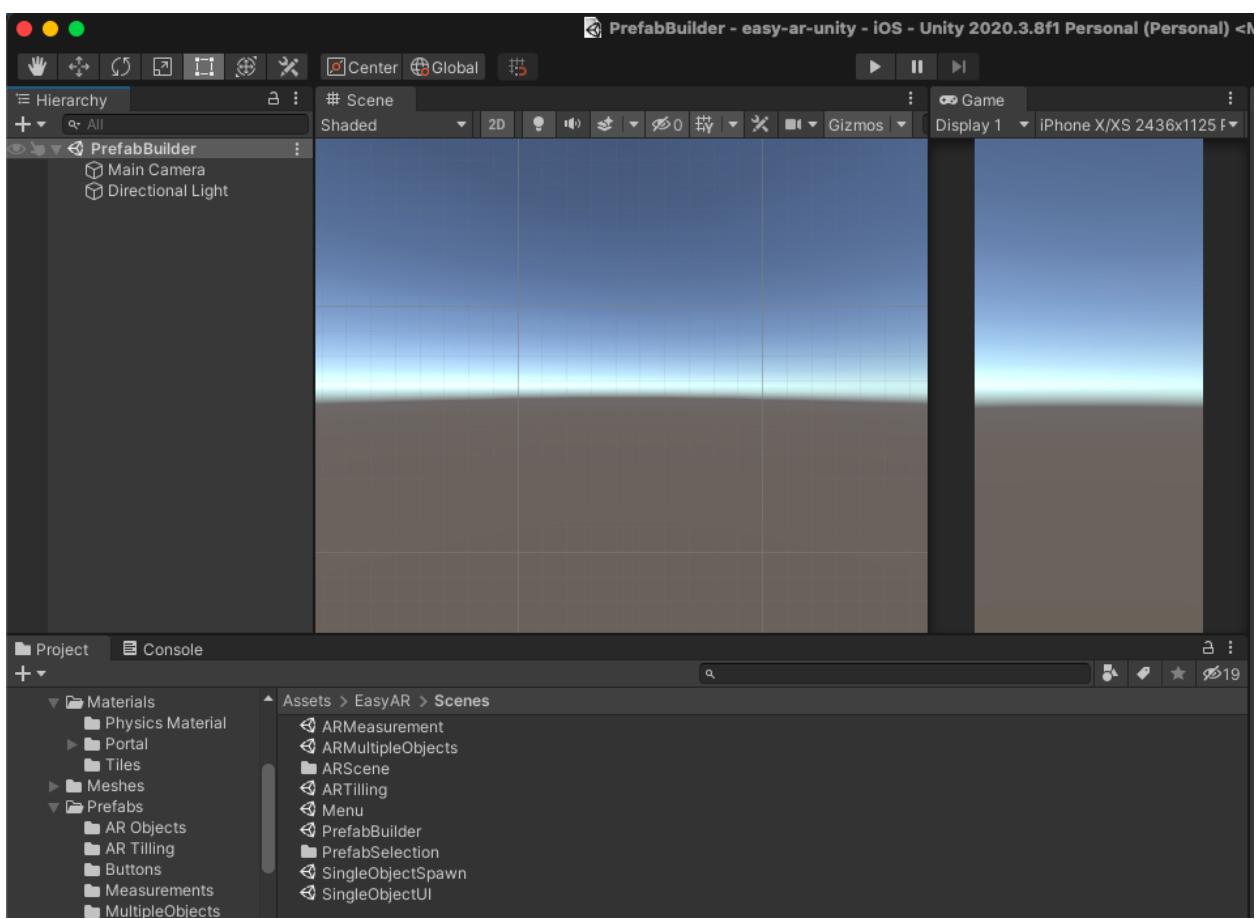


5.7. Here we go...now you can build the app by pressing the build button in the build setting.

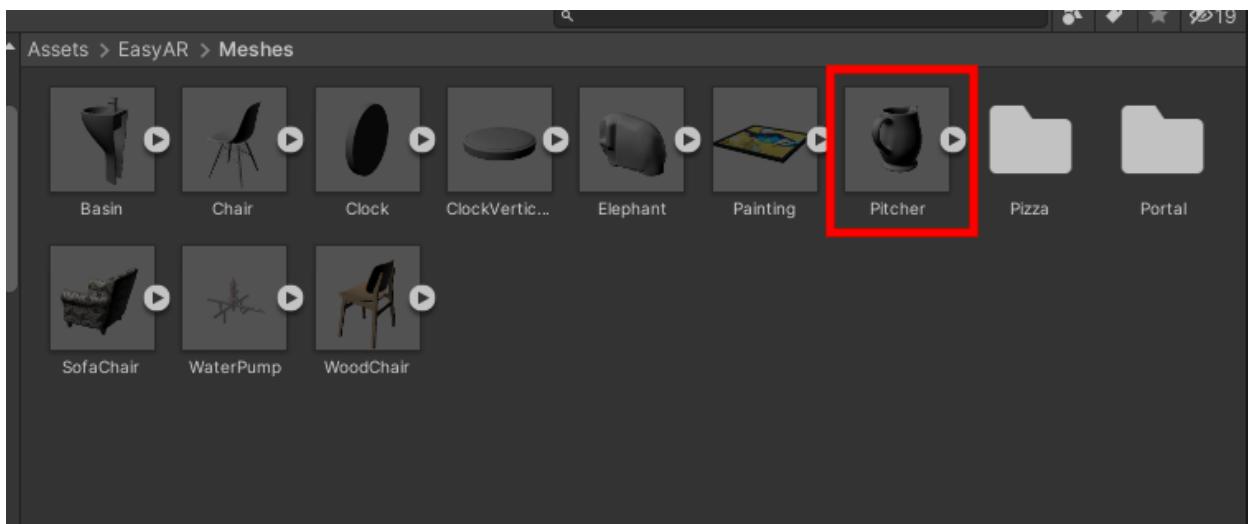
6. How to create a new object prefab

Easy AR already included prefabs in the [EasyAR->Prefabs](#) folder. All these prefabs are made in a specific structure. From this section we guide you to create your own 3d object prefab.

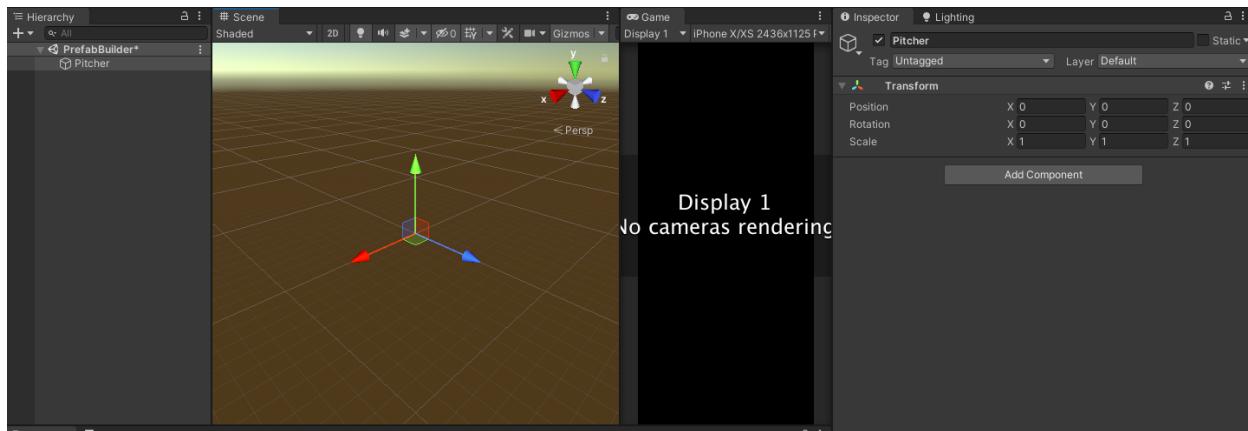
- **Step 1** : Create a new scene named “[PrefabBuilder](#)” or desired name you need in [EasyAR->Scenes](#) folder & load the scene.



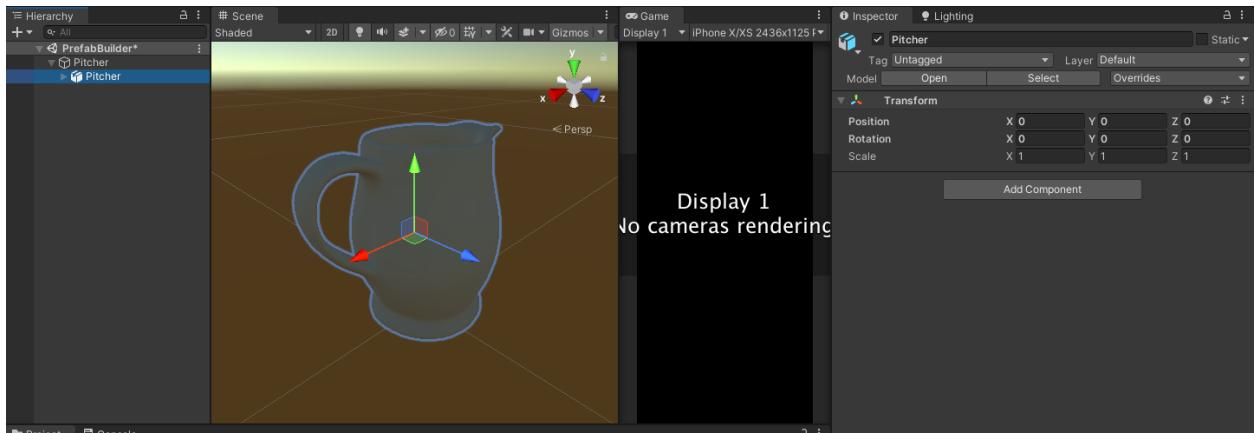
- **Step 2** : Delete Main Camera & Directional light from hierarchy.
- **Step 3** : Add the 3d object that you wish to create a prefab from [EasyAR->Meshes](#) folder. (for example we are going to make a prefab from our **pitcher** model)



- **Step 4** : Create an empty game object & name it as 3dmodel name itself. (for example we name it “**pitcher**”). Reset the transform of that game object.

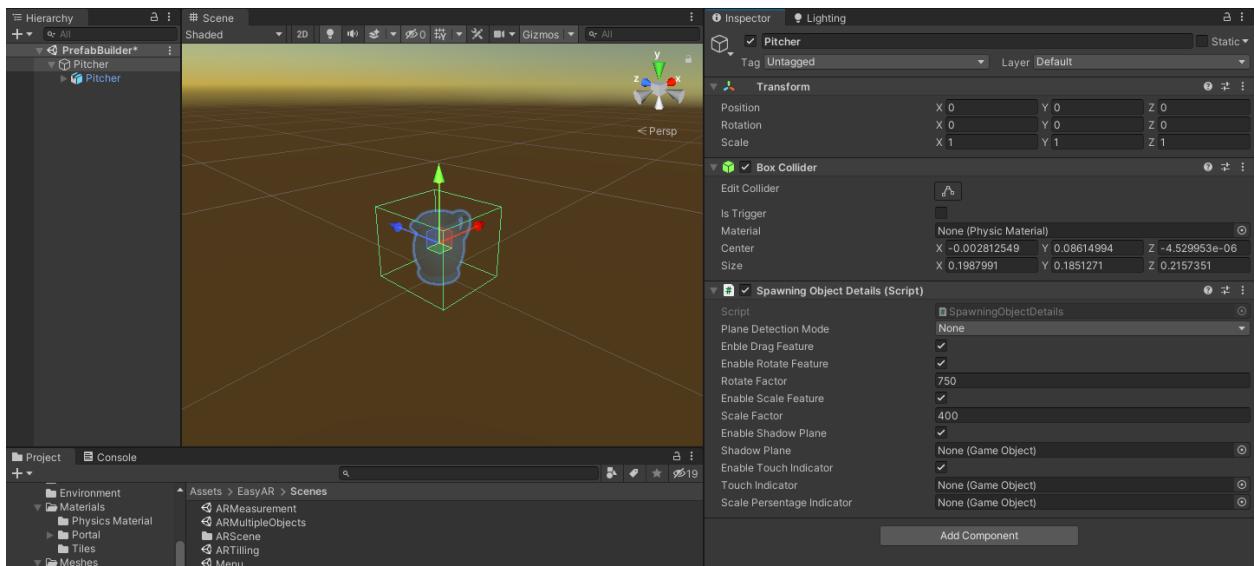


- **Step 5:** Drag & drop the 3d model into the hierarchy window as a child of the empty game object. (For example we select the pitcher 3d model). Reset the transform of the 3d model. Sametime you can add the relevant materials to your object.

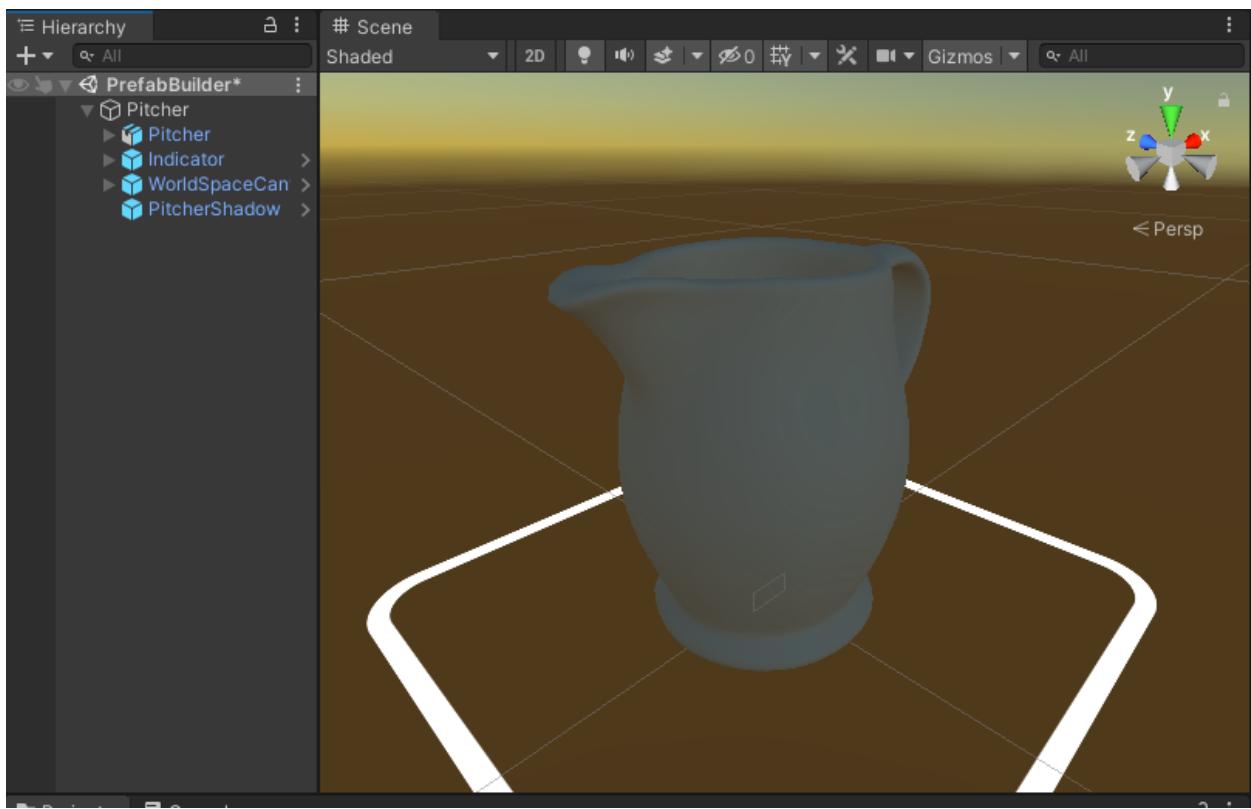


(Please consider the size of the 3d object can be changed as you desire. Also if you create a vertical placement object make sure object face towards Y up direction)

- **Step 6** : Select empty game object which was created before & go to inspector. Add box collider to it & set size same as game object's (pitcher) size. Also add `SpawningObjectDetails` script from `EasyAR->Common->Scripts` to it.

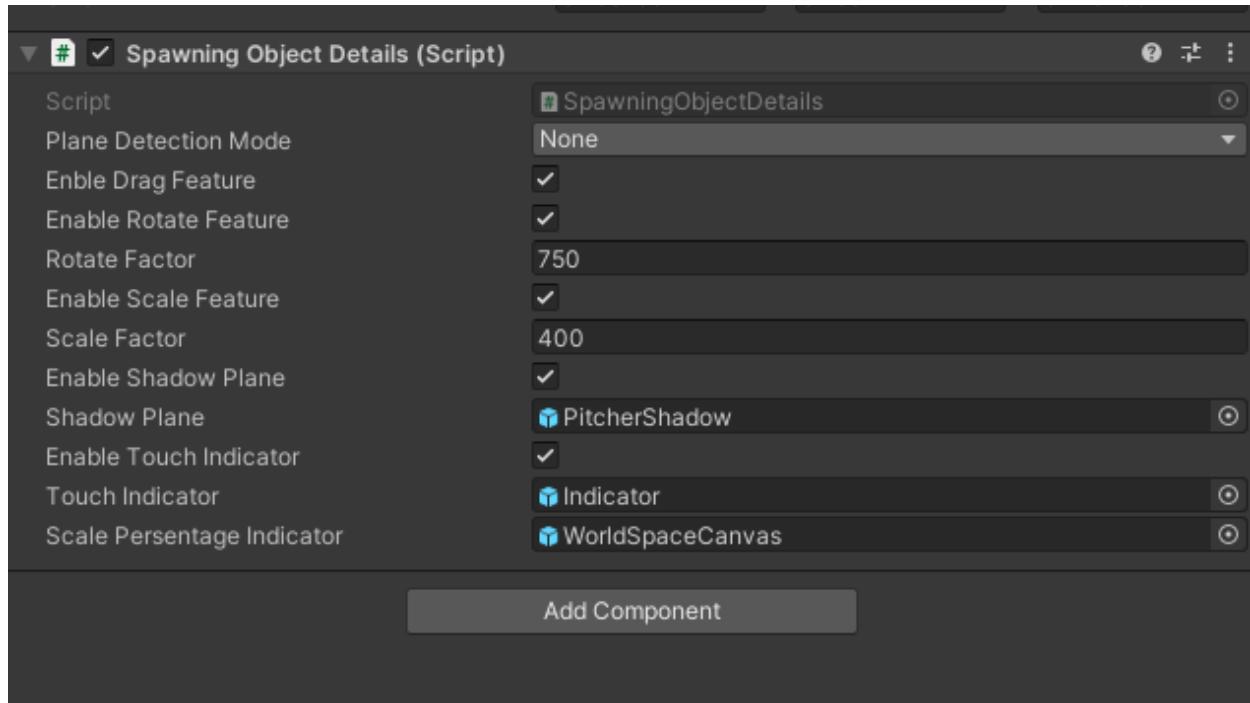


- **Step 7** : Drag & drop the following game objects under the empty gameobject respectively as child objects.
 - Indicator prefab (**EasyAR->Prefabs**)
 - Adjust the Indicator (position, rotation and scale) accordingly to the object vertical or horizontal placement.
 - WorldSpaceCanvas Prefab (**EasyAR->Prefabs**)
 - ShadowPlane Prefab (Goto [Section 8](#) to see how to generate the shadow plane)



- **Step 8** : Select Empty object (pitcher) & go to Inspector. Under the **SpawningObjectDetails** script you have to drag & drop below three prefabs accordingly.
 - For **Shadow Plane**, drag & drop **gameobject's shadow prefab** (pitcher shadow) from hierarchy.
 - For **Touch Indicator**, drag & drop **Indicator prefab** from hierarchy.

- For **Scale Percentage Indicator**, drag & drop the **WorldSpaceCanvas** prefab from hierarchy.



- **Step 9:** Set the plane detection mode in the Spawning object details component into **Vertical** or **Horizontal**. (Do not set it as **Everything**)
- **Step 10:** To create the prefab you can go to the **Easy AR->Prefabs** folder and drag and drop the Empty object (pitcher) to this folder.

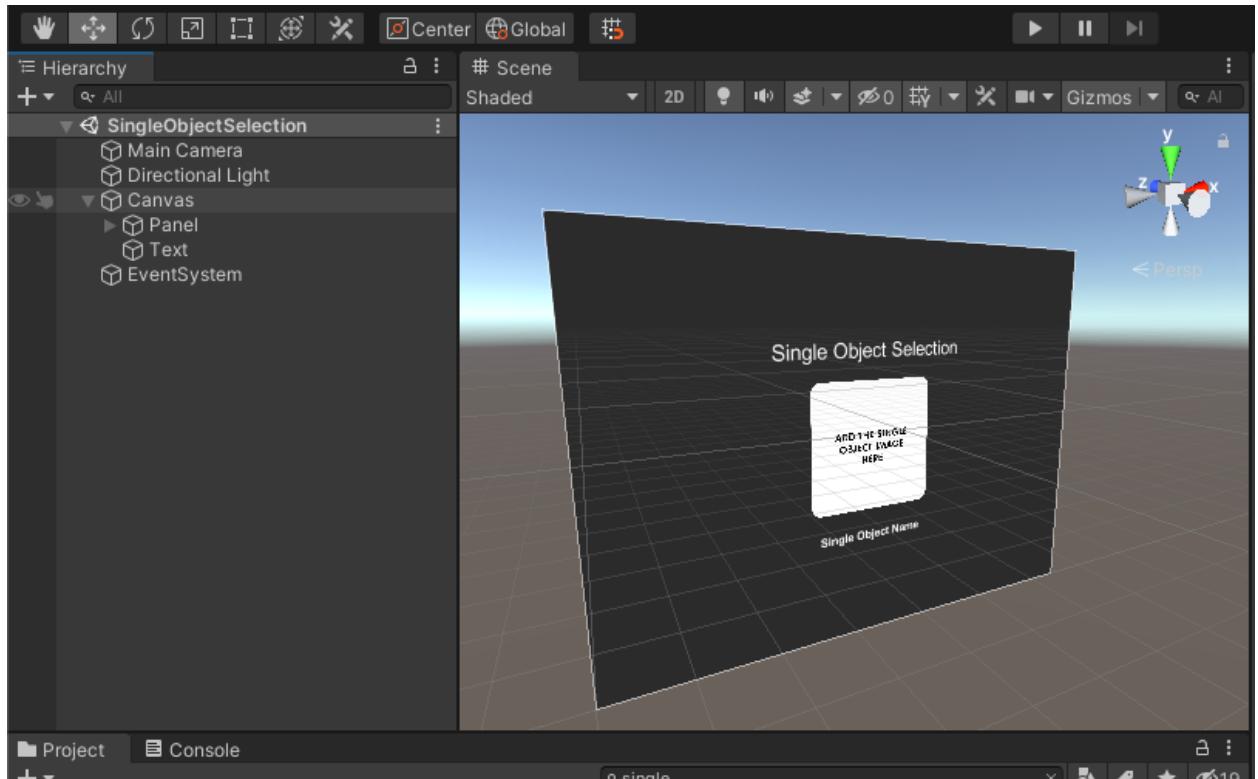
You are done...Now you have created your own prefab to be spawned.

7. Setting up a custom scene

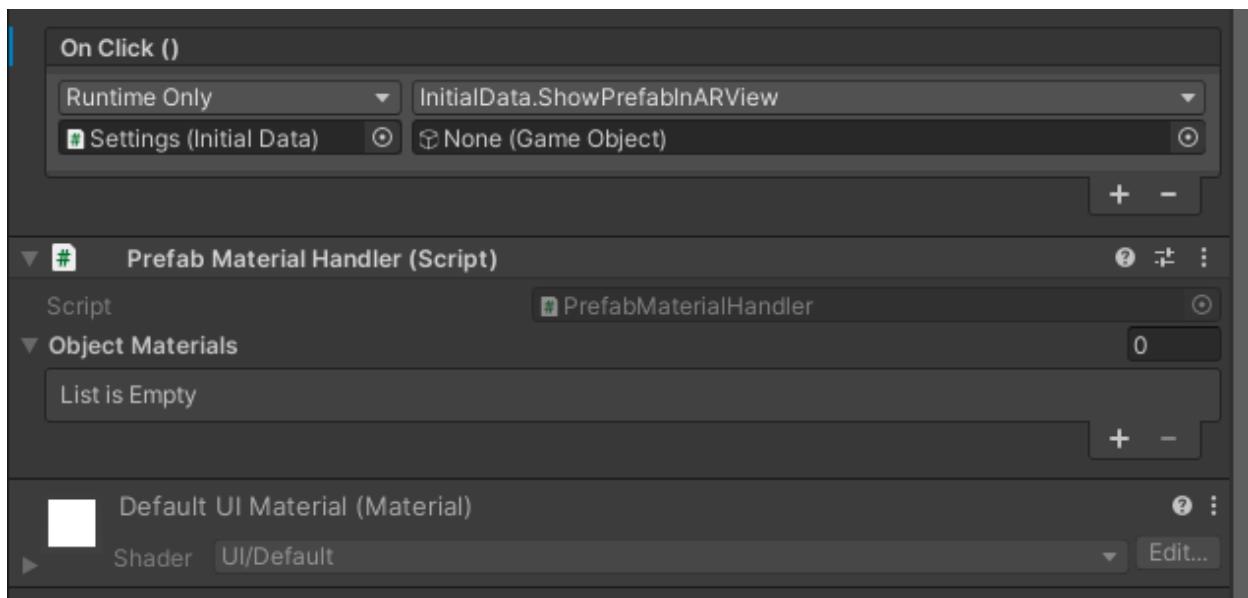
7.1. How to set up a single object placement

This will guide you how to add a single object with easy manipulation (rotation/scaling/change position with shadow) into your project.

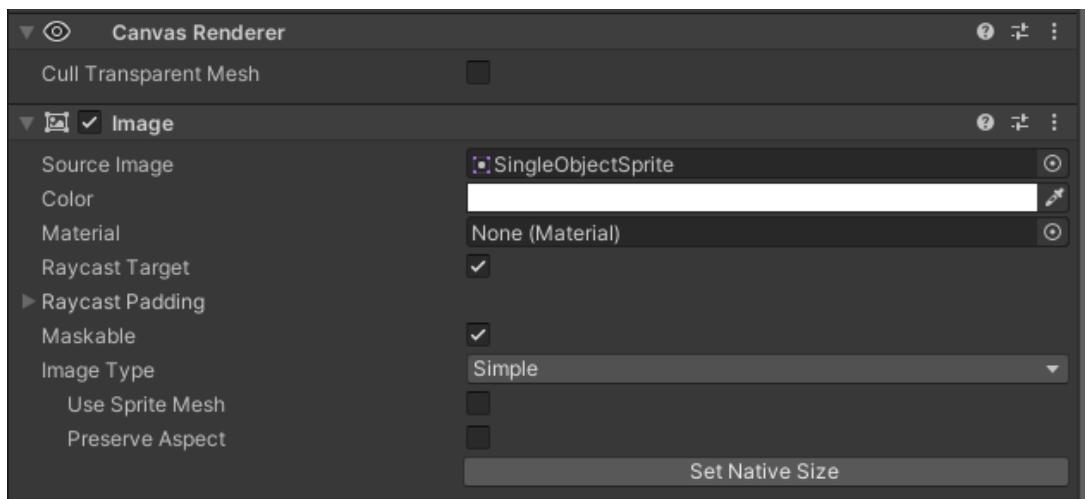
- **Step 1 :** Go to the [EasyAR->Scenes](#) folder & load the **SingleObjectSelection** scene.



- **Step 2 :** Under the Canvas, a Panel has been added with **singleObjectSelectionButton** prefab. Select this prefab and in the inspector you would see an empty field in the Button component **OnClick()** event named as **None(Game Object)**.

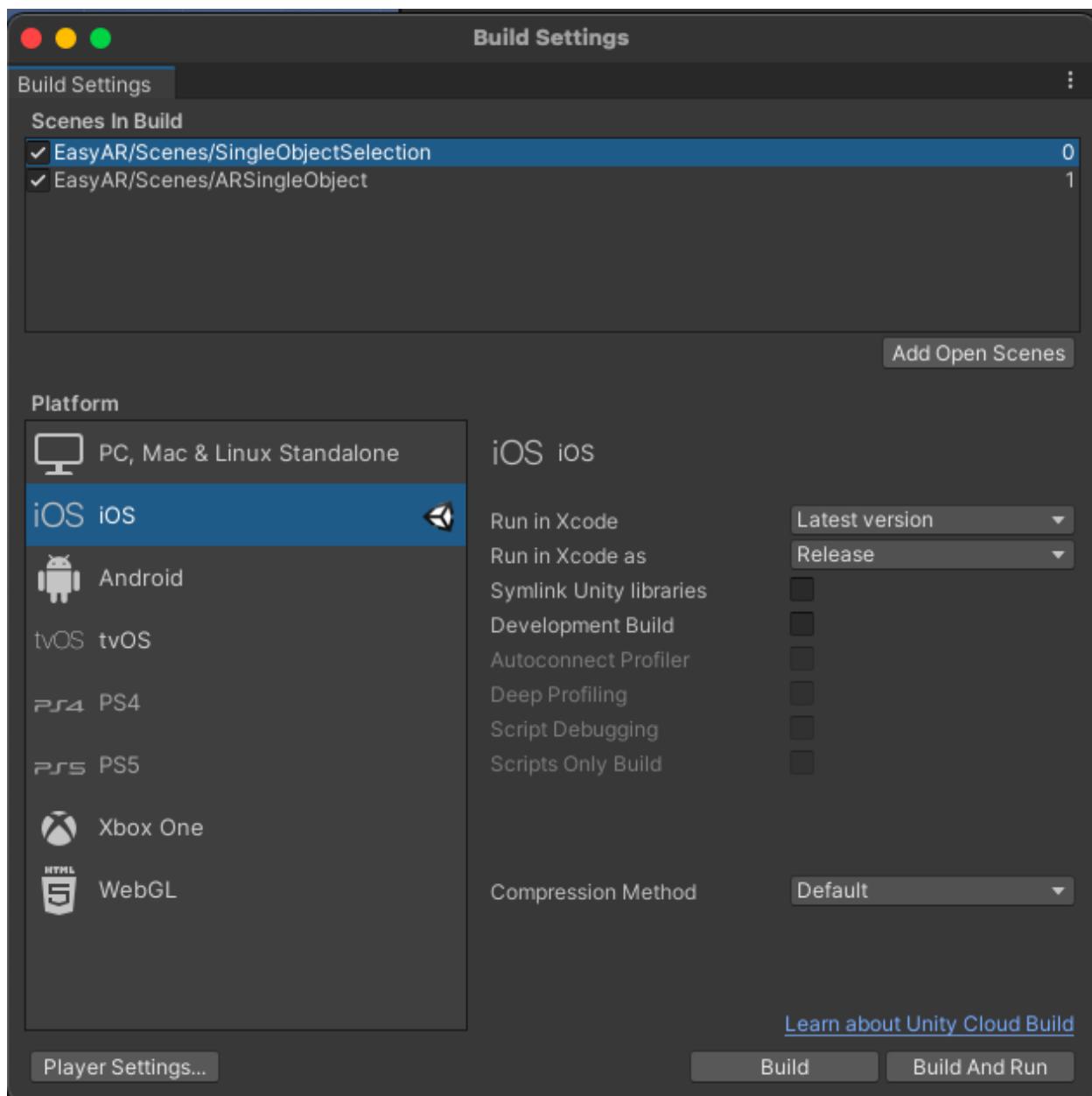


- **Step 3 :** Drag and drop your preferred created prefab (as mentioned in [Section 6](#)) from the `EasyAR->Prefabs` folder to this **None(Game Object)** area.
- **Step 4 :** At the same time you can drag and drop the relevant materials of the prefab you created to the Object Materials in the **Prefab Material Handler** script.
- **Step 5 :** If needed, you can change the sprite of the button prefab with a preferred relevant name



(You can add any number of **singleObjectSelectionButton** prefabs as you need in this scene)

- **Step 6:** Now you are ready to build the scene with your custom prefab.. Go to build setting & Remove all the scenes under scenes in the Build Settings. Then set the first scene as your **SingleObjectSelection** scene (the scene which is created now) & **ARSingleObject** scene from the EasyAR->Scenes folder respectively.

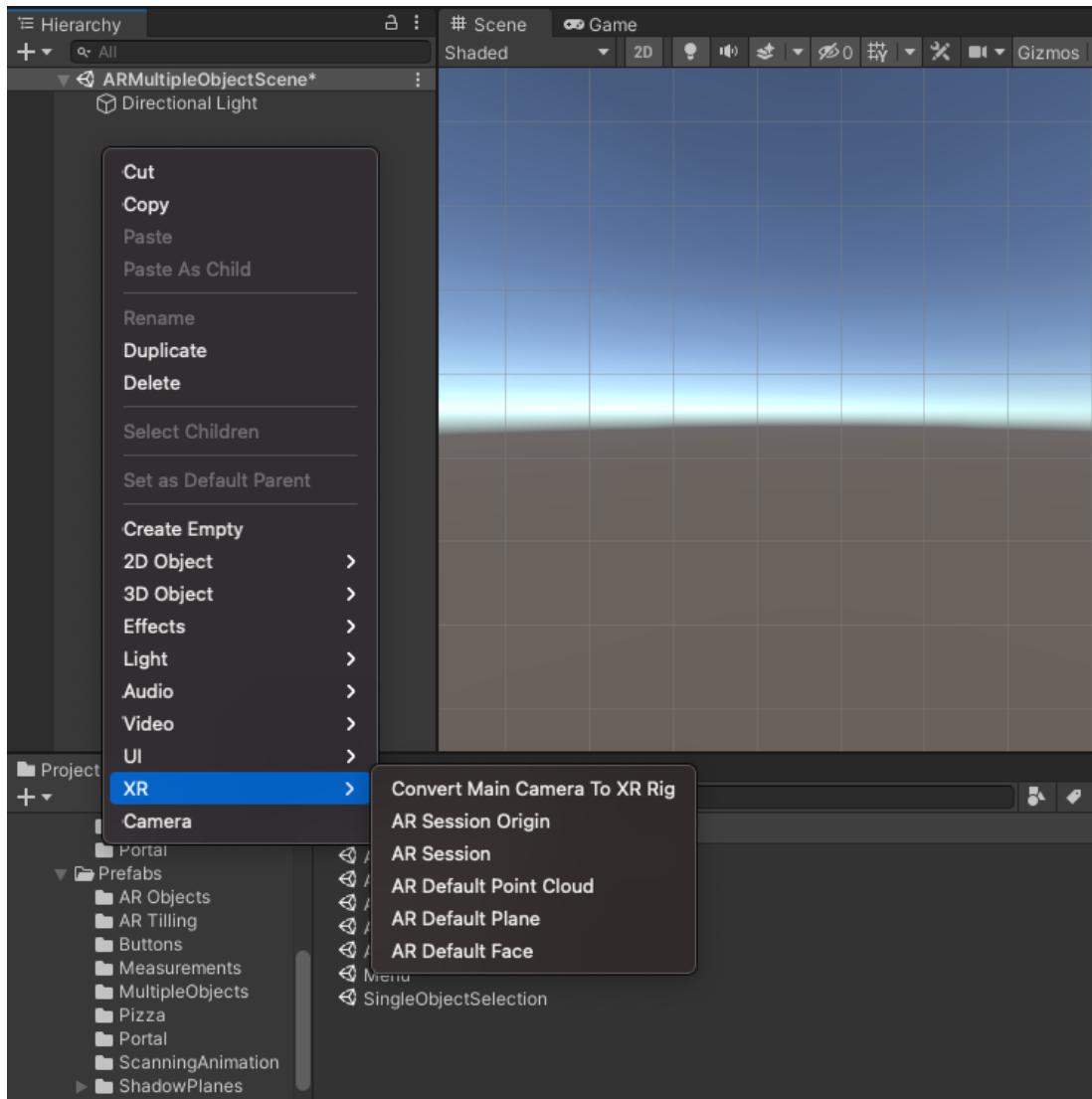


- **Step 7:** Select your desired platform & build. (Refer **section 5.4 - 5.7** to change the build setting according to the platform you choose). Enjoy the Easy AR app.

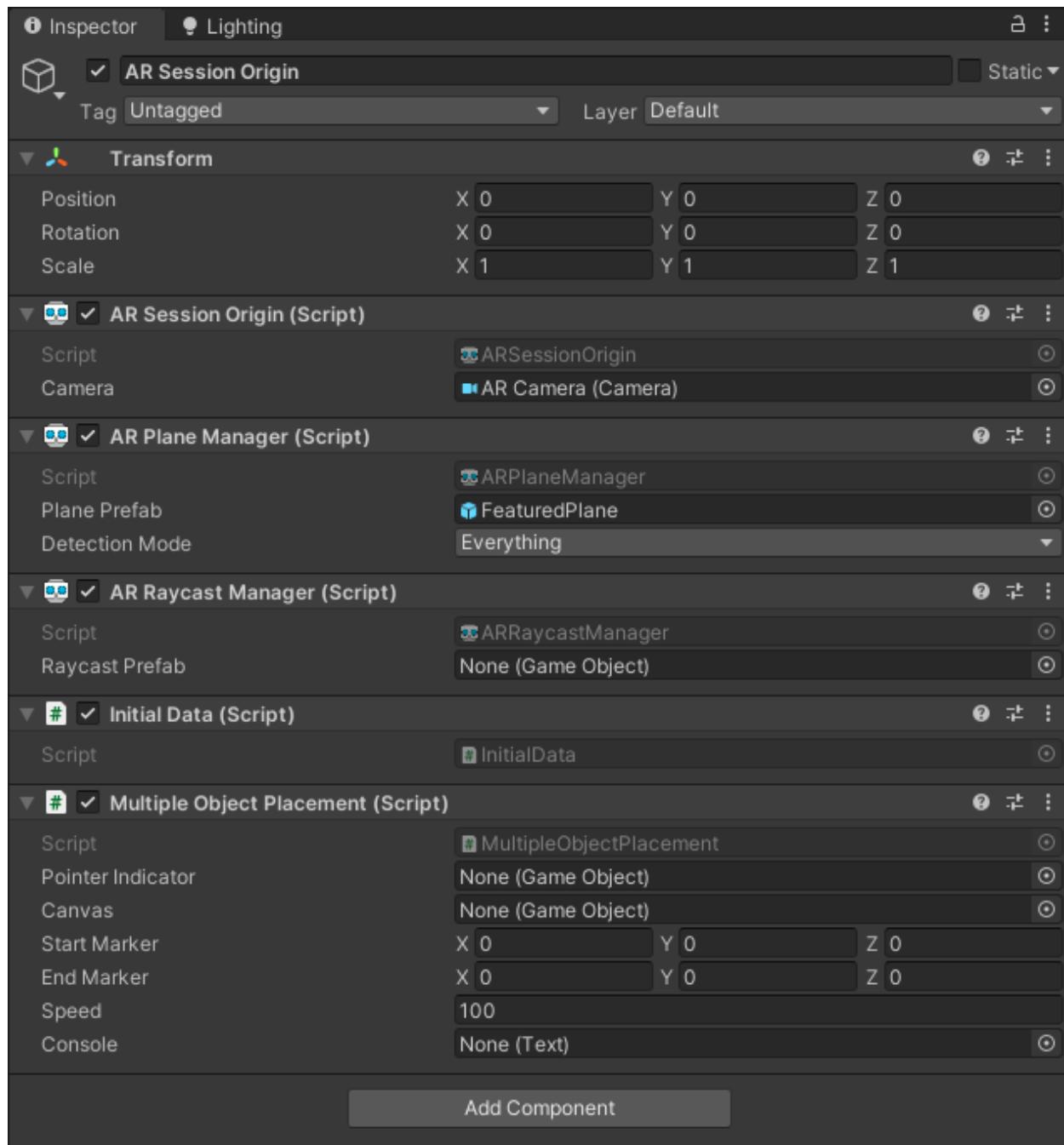
7.2. How to set up the Multiple Object Placement

In this section, we will guide you to how to create ARMultipleObjectScene.

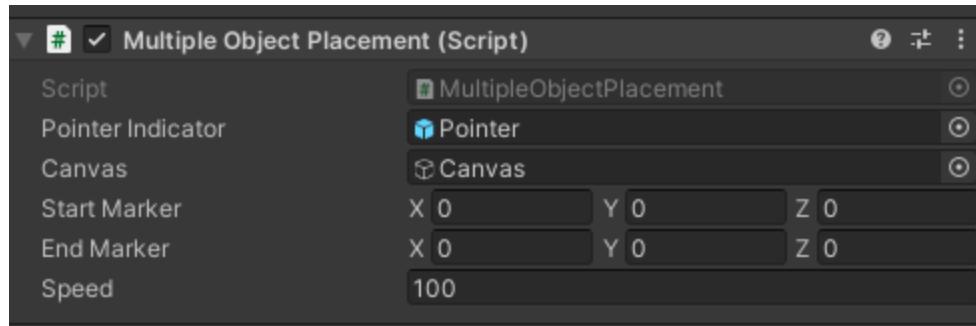
- **Step 1:** Create a new scene and name it as “[ARMultipleObjectScene](#)” or any preferred name.
- **Step 2 :** Delete Main camera from scene and Add AR Session, AR Session Origin from right click on the hierarchy and goto XR then select AR Session and AR Session Origin



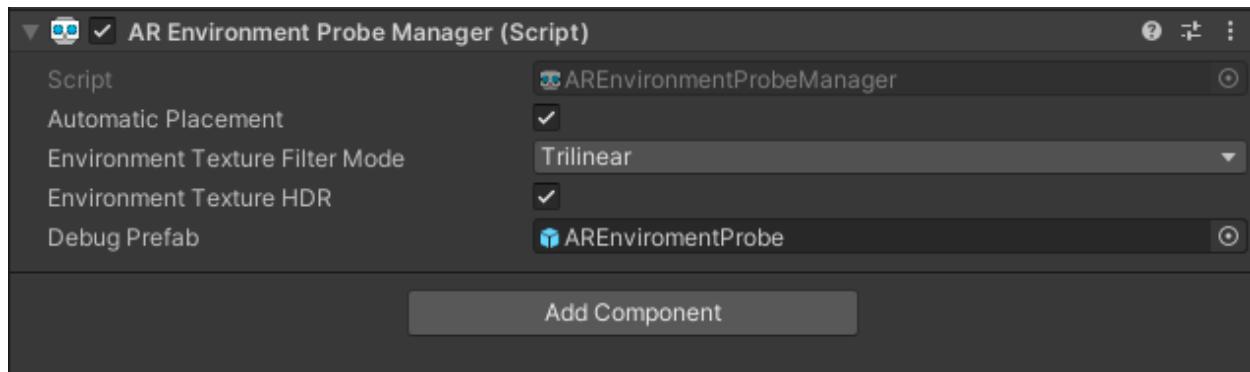
- **Step3** : Select AR Session Origin & go to inspector. Add **MultipleObjectPlacement** script from **EasyAR->Scripts** folder. Also add **ARPlaneManager** component to AR Session Origin gameobject.



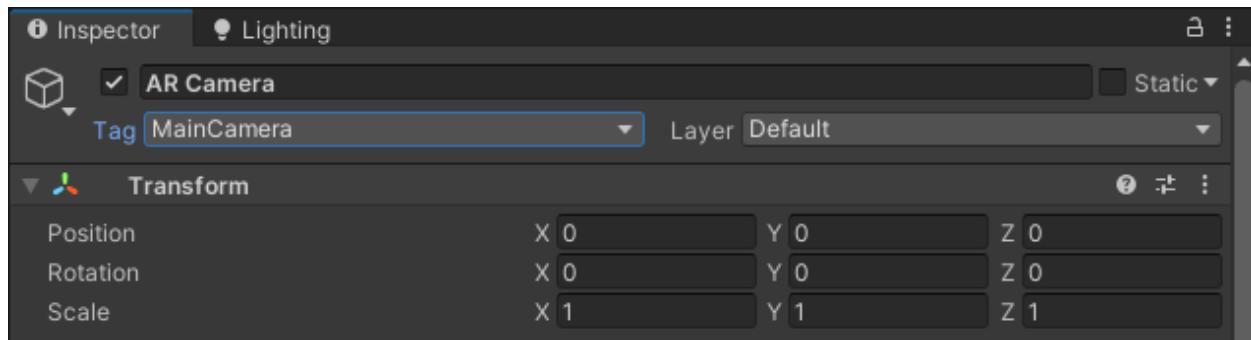
- **Step 4:** For a pointer indicator you can add the [pointer prefab](#) from the [EasyAR->Prefabs](#) folder. For Canvas you need to create a new canvas in game hierarchy and drag & drop in to the Canvas game object in [Multiple Object Placement](#) component.



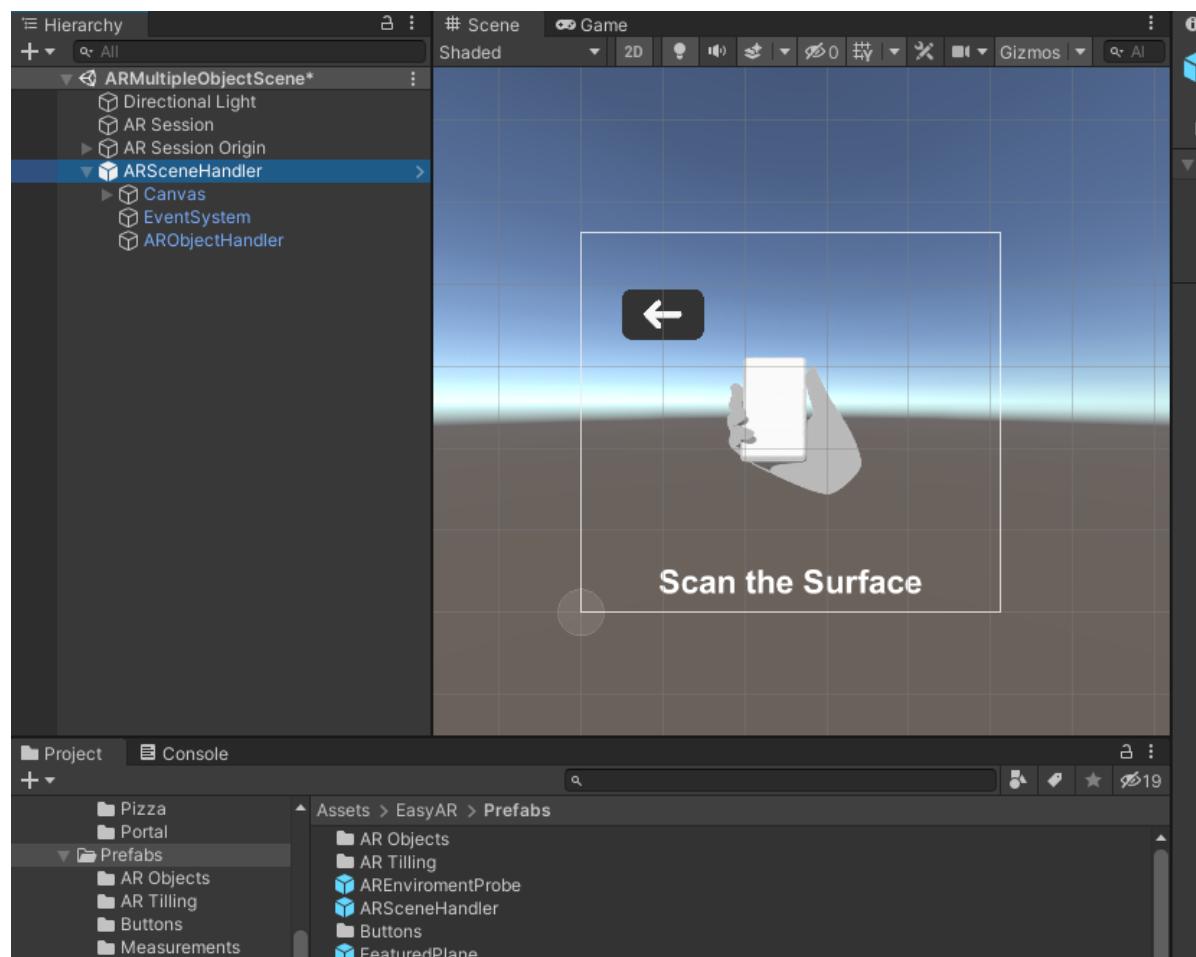
- **Step 5:** Also you need to add AR Environment Probe Manager component to AR Session Origin gameobject. For debug prefab drag and drop **AREnvironmentProbe** prefab from [EasyAR->Prefabs](#) folder.



- **Step 6:** Select AR Camera under AR Session Origin gameobject & Change the Tag of the AR Camera as Main Camera

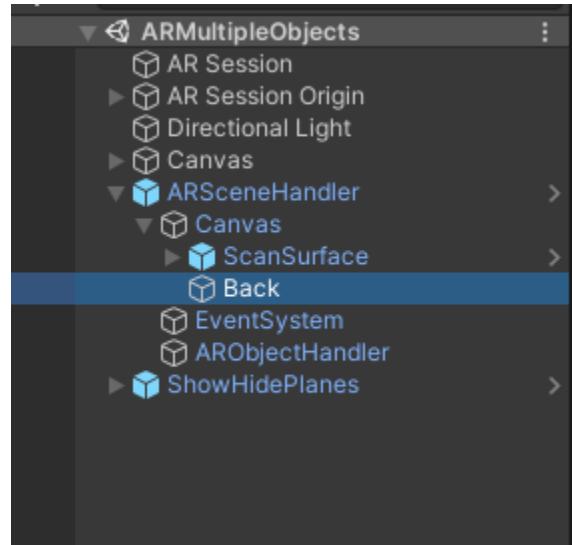


- **Step 7 :** Add ARSceneHandler prefab from EasyAR->Prefabs folder to hierarchy.

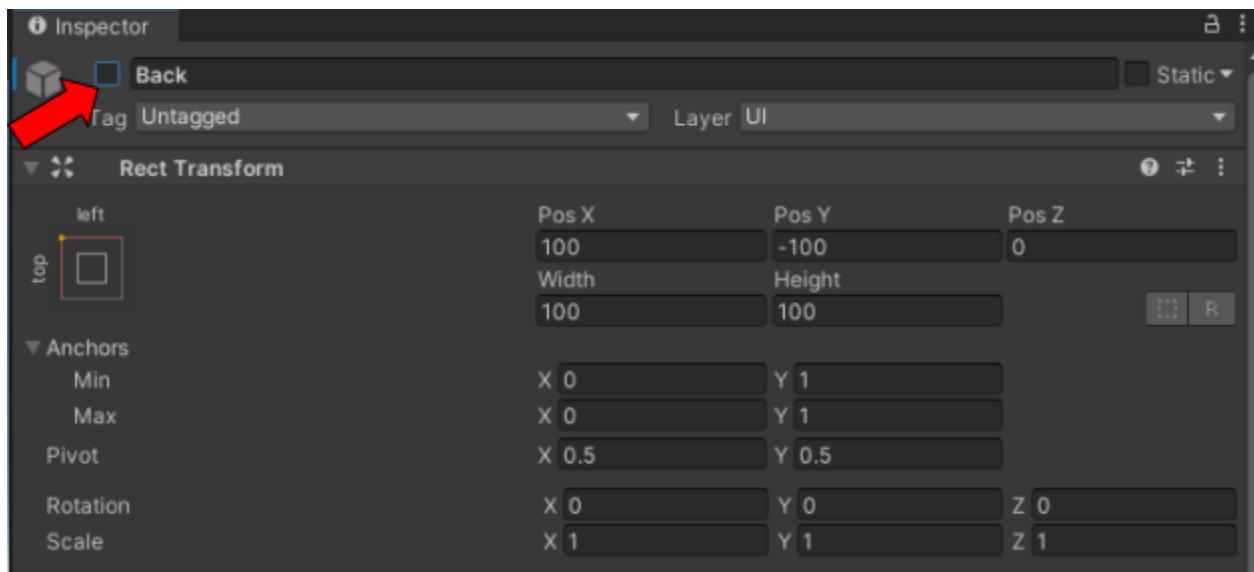


- **Step8:** If you don't need a back button you can hide it.

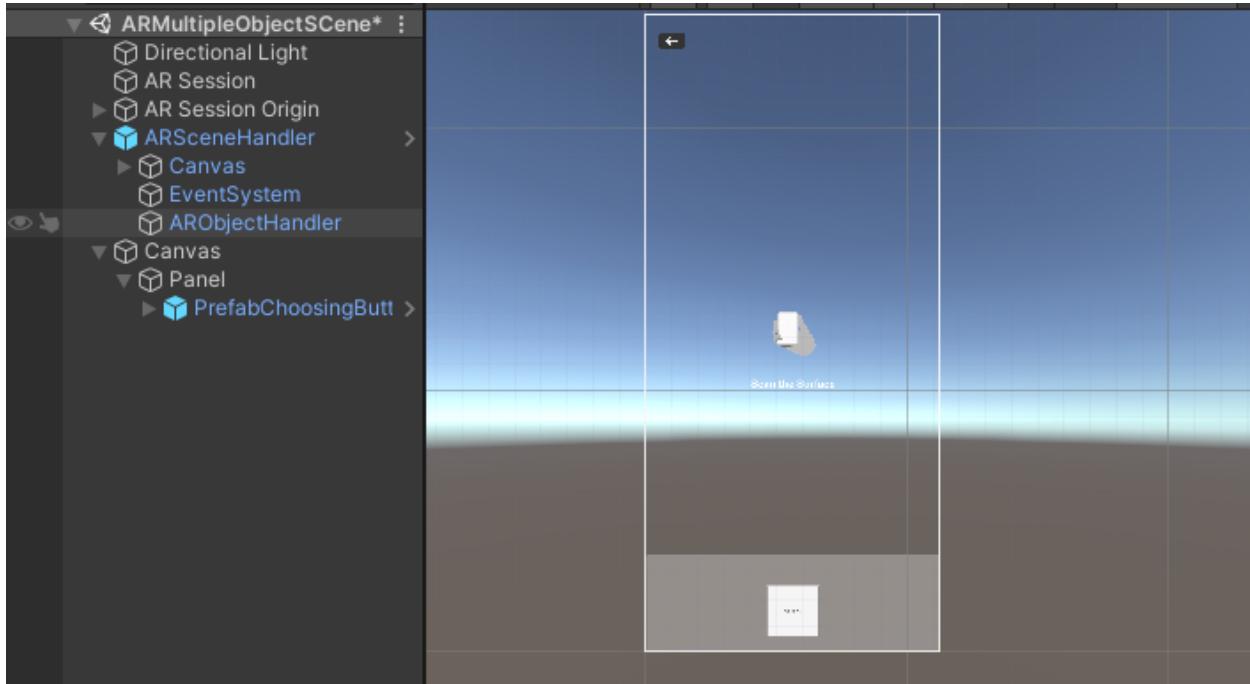
- Select back button from hierarchy.



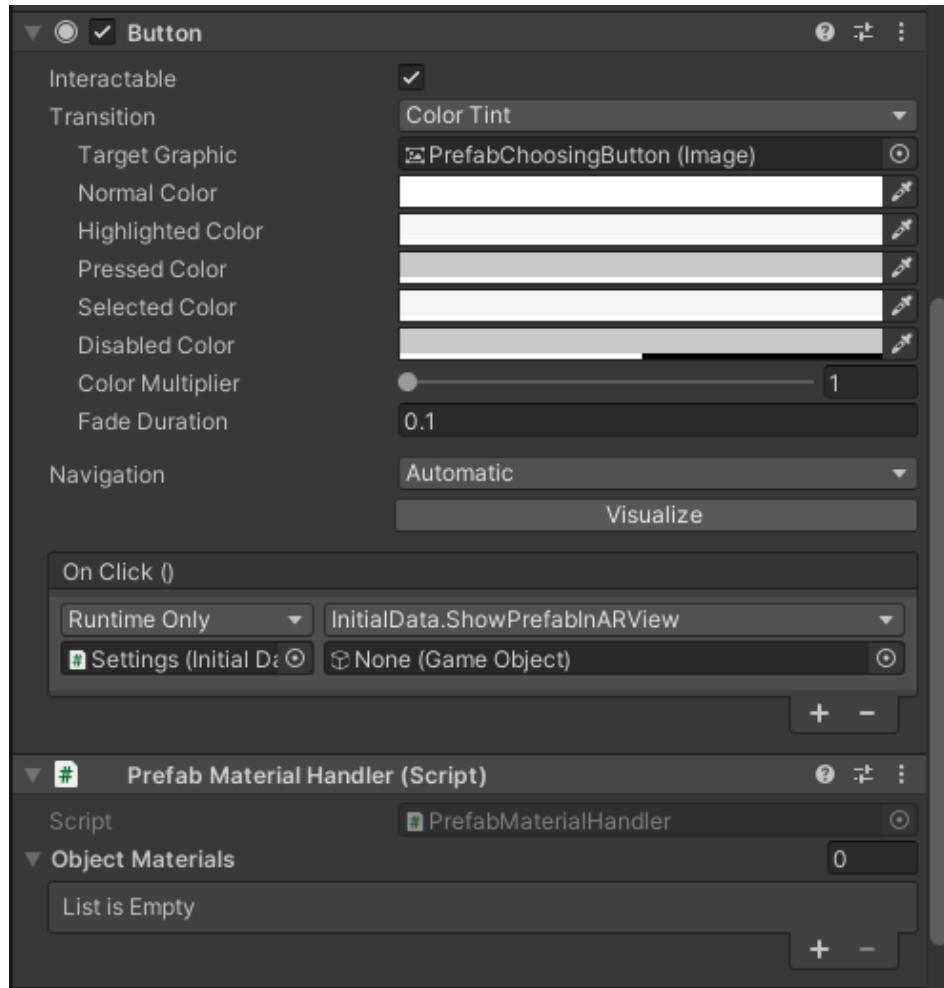
- Untick the game object from the inspector.



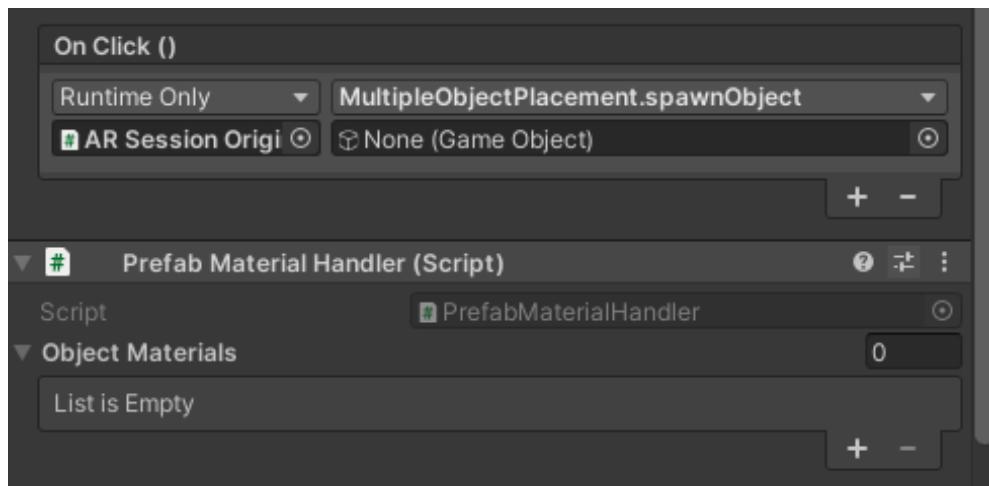
- **Step 9:** In hierarchy goto newly created Canvas & Add new panel. Add `prefabChoosingButton` prefab from `EasyAR->Prefabs->Buttons` as a child object in the panel.



- **Step 10:** This one `PrefabChoosingButton` prefab is responsible for showing one AR object. You can add multiple `PrefabChoosingButton` prefabs as you need as a child object of the panel. Click `PrefabChoosingButton` & go to the inspector.

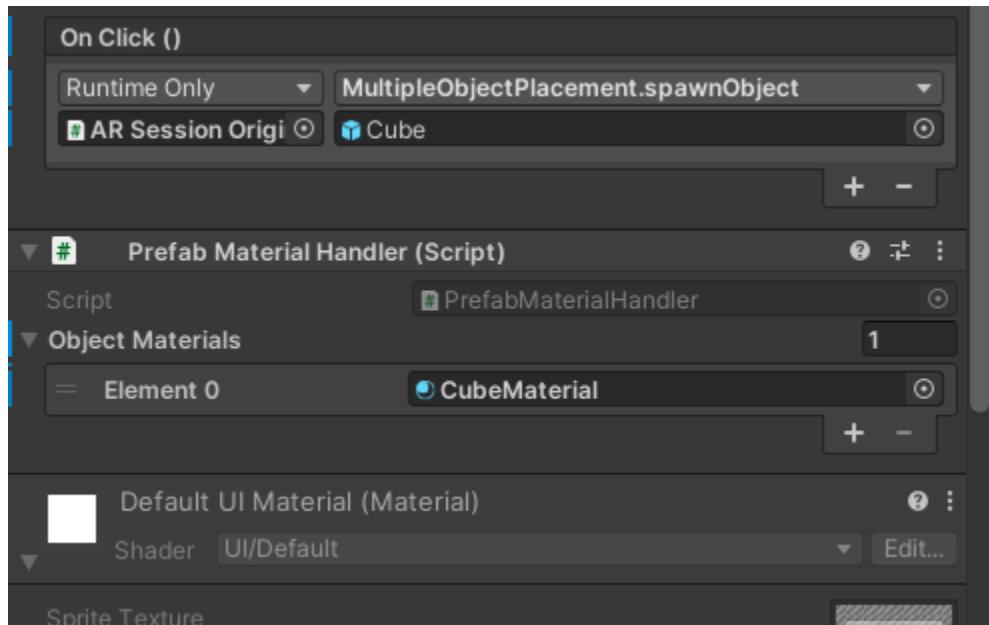


- **Step 11:** Under the Onclick function it has already added Settings gameobject. Remove it & Add new -> drag & drop AR Session Origin game object from Hierarchy.

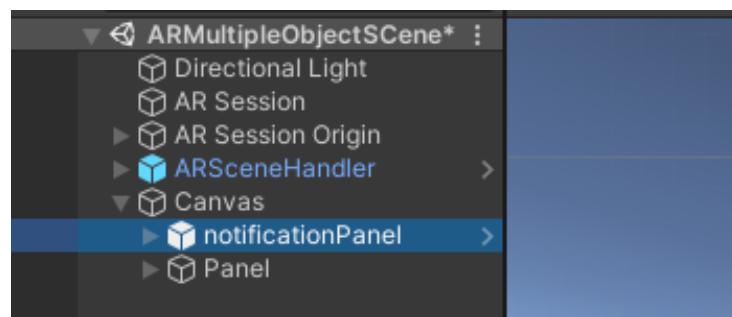


- **Step 12:** Select `MultipleObjectPlacement` script & `spawnObject` method. You can drag & drop the game object prefab that you need to spawn here. Same time you can add a number of prefab materials & add materials to list in `Prefab Material Handler Script`.

Make sure when you create a Spawning object prefab should follow the same steps in steps followed at single object placement developed. (Refer step 7-10 in section 6.1)



- **Step 13:** Navigate to Canvas again in hierarchy. Add a [notification panel](#) as a child of canvas. You can find the [notification panel](#) prefab from the [EasyAR->Prefabs](#) folder. This panel is responsible for showing error messages if a user tries to place a vertical placement object in Horizontal planes or horizontal placement objects placed in a vertical plane.



- **Step 14:** Yes you reach the final step. Go to the build setting & choose your preferred platform & build.(Add the scene to build settings if the scene is not in build setting)

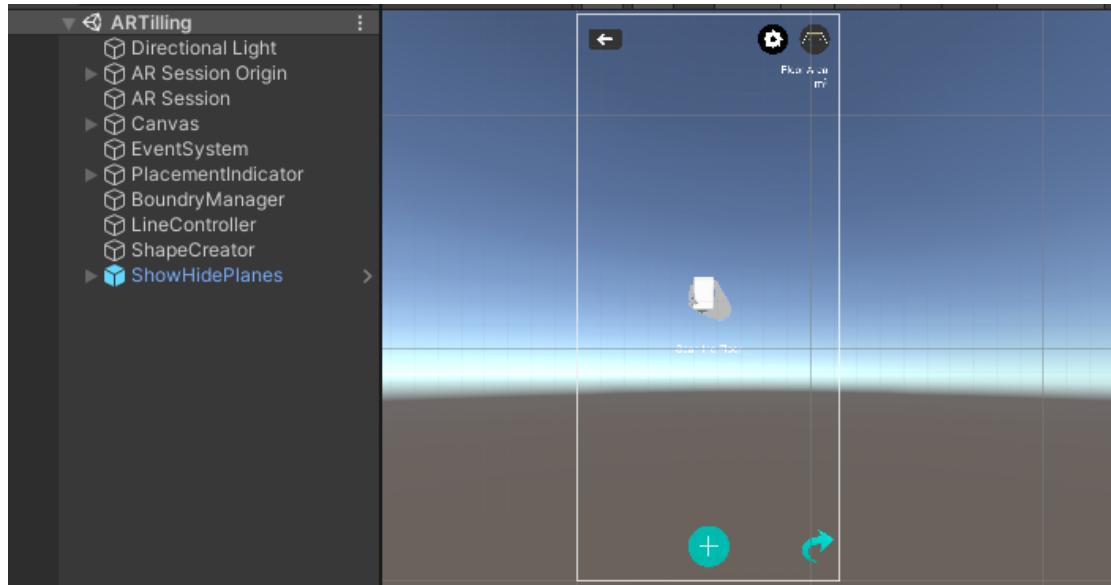
Things you need to consider before building the project.

- ❖ Make sure AR camera tag name set as **Main camera**
- ❖ Make sure you add the notification panel & tag name is set as **NotificationPanel**.
- ❖ Make sure all spawning objects prefabs include relevant gameobjects discussed in step 7-10 in section 6.1.
- ❖ Make sure all spawning objects include colliders

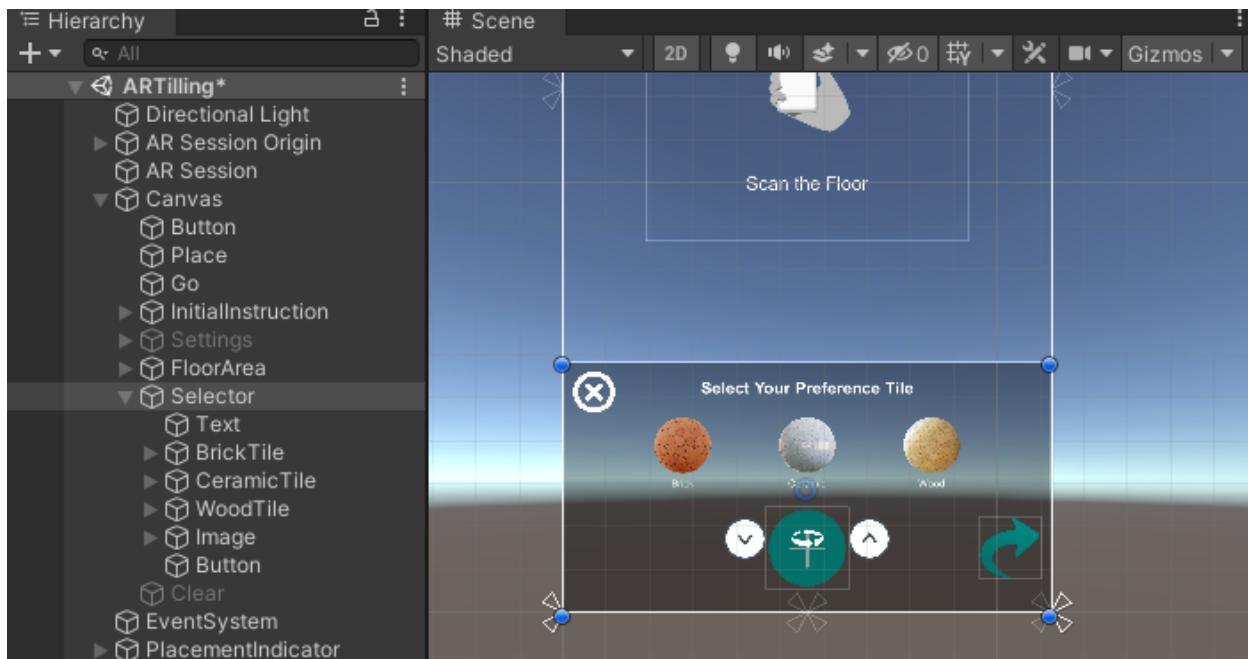
6.3. How to set up the AR tilling

For AR Tilling we have given you a demo scene named **AR Tilling**. The scene can be found in the **EasyAR->Scenes** folder. From here we are going to add or modify existing tiles in the scene.

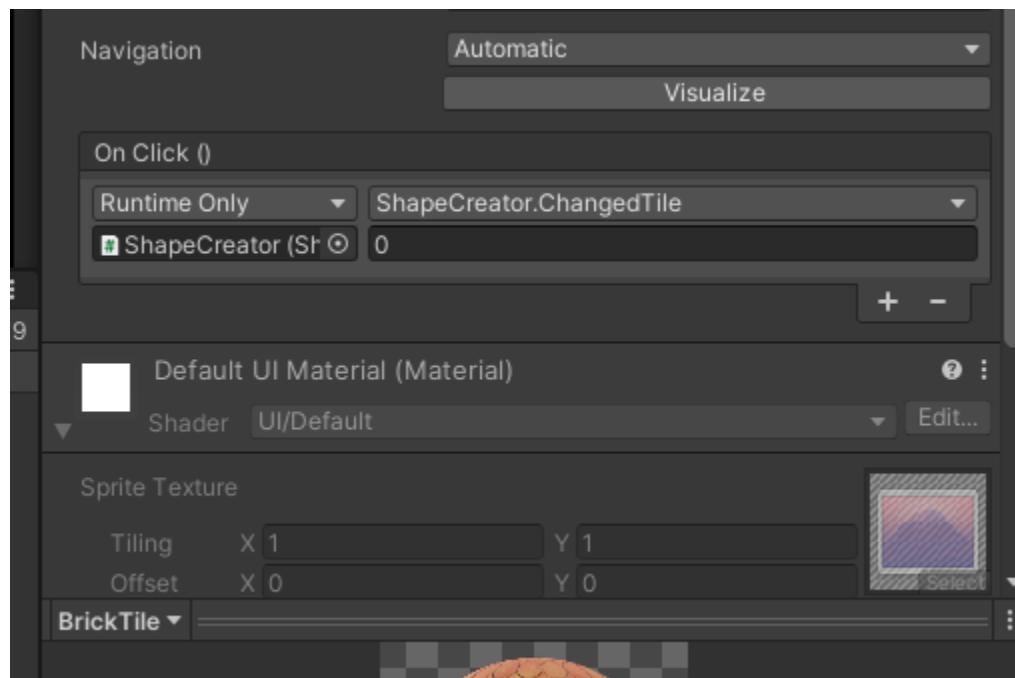
- First you need to open the scene named **AR Tiling**. You can see the hierarchy shown below.



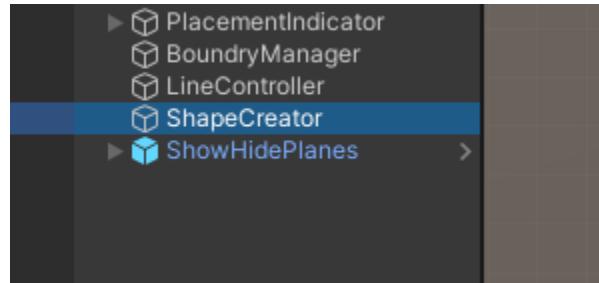
- The tile changing scenario happens inside the canvas. Expand the canvas & enable selector gameobject (by default this gameobject disabled). Now you can expand the selector also.



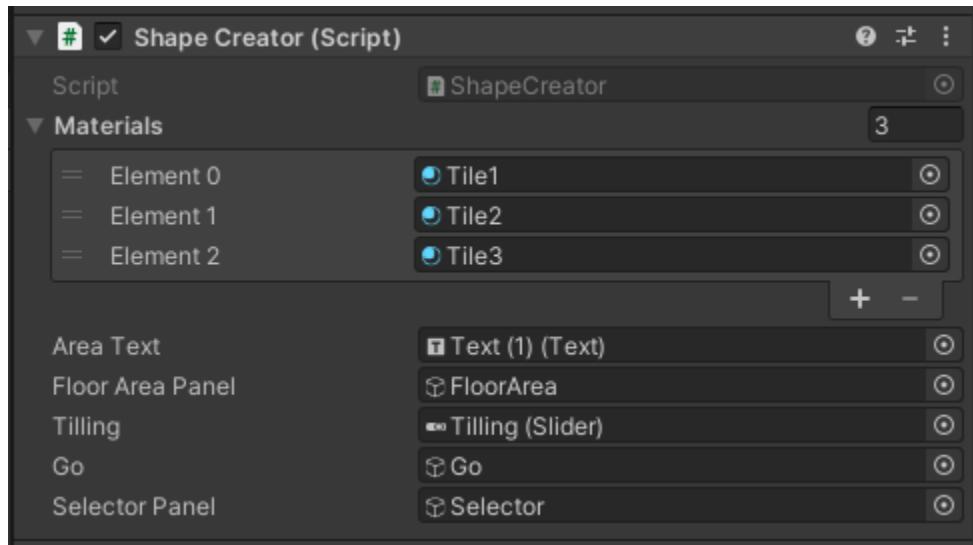
- As an example shows we added 3 different tiles named Brick Tile, Ceramic Tile & Wood Tile. These are simply buttons. Click one button & check the inspector.



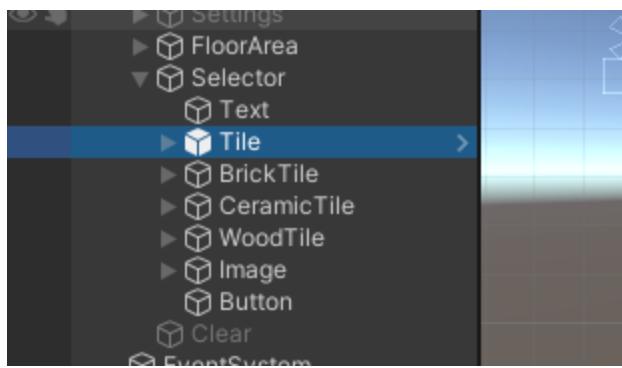
- You can see, the button onclick function called shape creator game object's change tile method. Shape creator gameobject can be found in the game hierarchy.



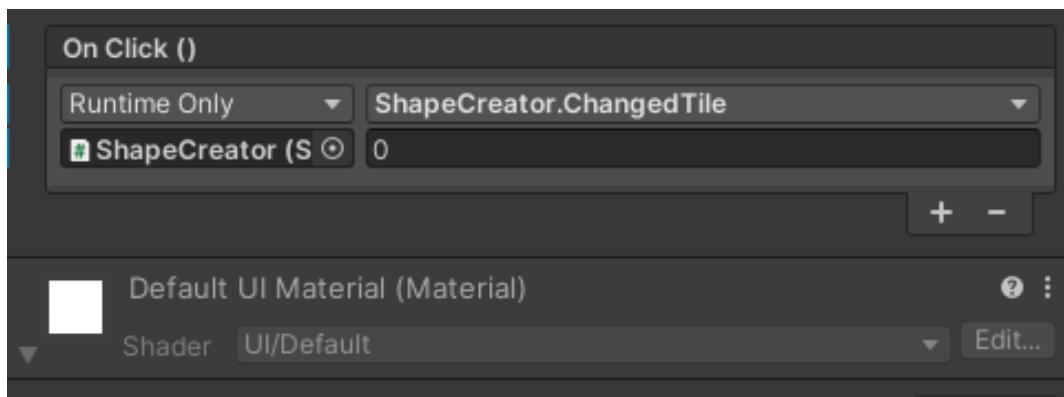
- Click shape creator gameobject & see inspector.



- You can see in **Shape Creator** we added 3 tile materials. If you need to modify material, simply you can change the material you need here. If you add new material, set materials count & set the materials.
- If you add new materials, you should have the same count of buttons in the **selector**. For custom buttons, you can drag & drop the **tile prefab** as a child of the selector from the EasyAR->Prefabs folder.



- Then select the tile prefab. Go to the inspector & click + button under the onclick method. Drag & drop shape creator game object to it & set ShapeCreator->changedTile method.

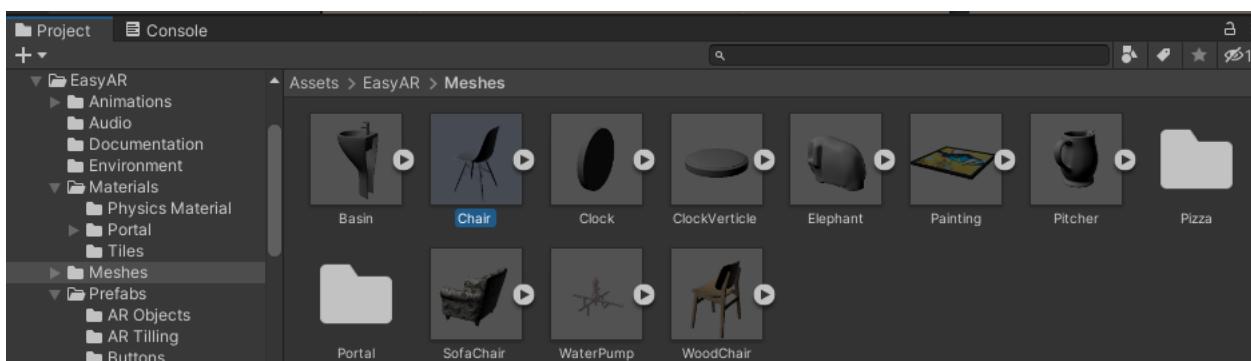


- Finally you can give the exact material element number here taken by the materials element in Shape Creator.
- You are Ready to build now..Go to the build setting. Change the build platform & build.

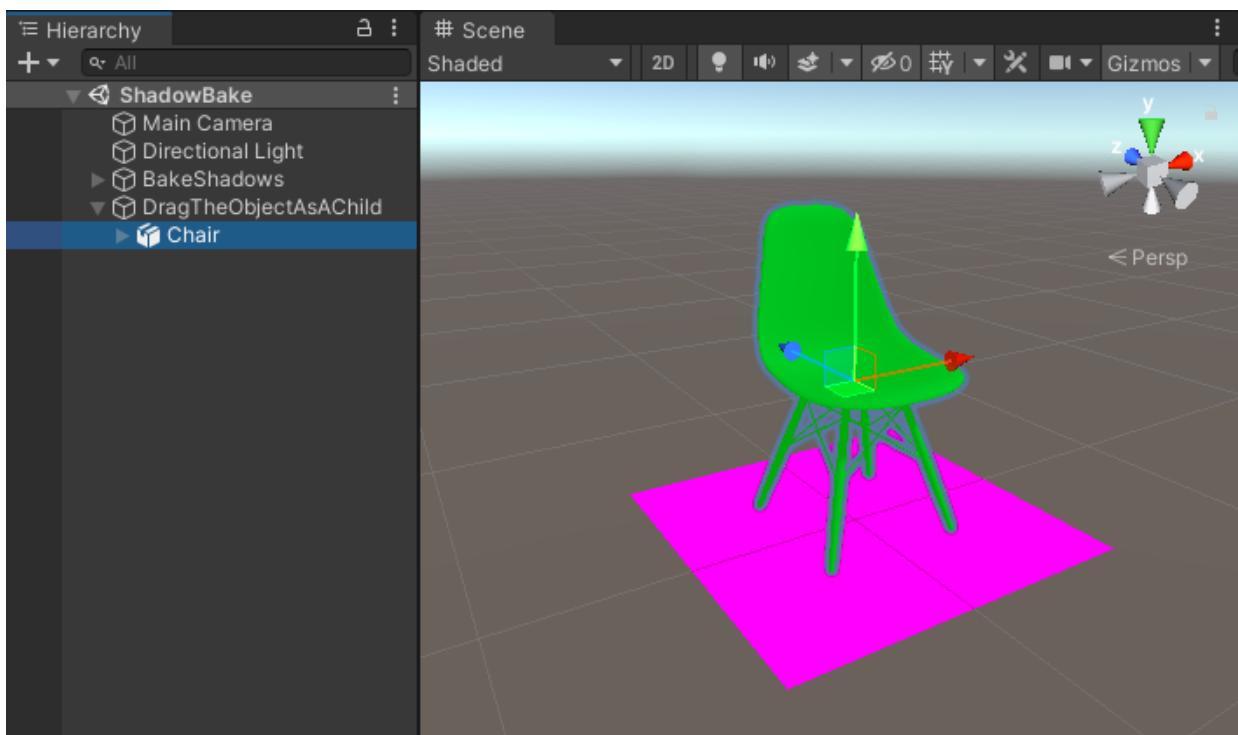
8. Shadow map generation

Easy AR gives you a new scene for shadow map generation of 3d objects in unity editor. We can use default shadows projection by directional light. But sometimes shadow took hard & random patterns. But after research we found a better way to implement a contact shadow map similar to IOS usdz shadow. So you can create your own smooth shadow map for 3d models using Easy AR. You are guided in this section, how to build this contact shadow.

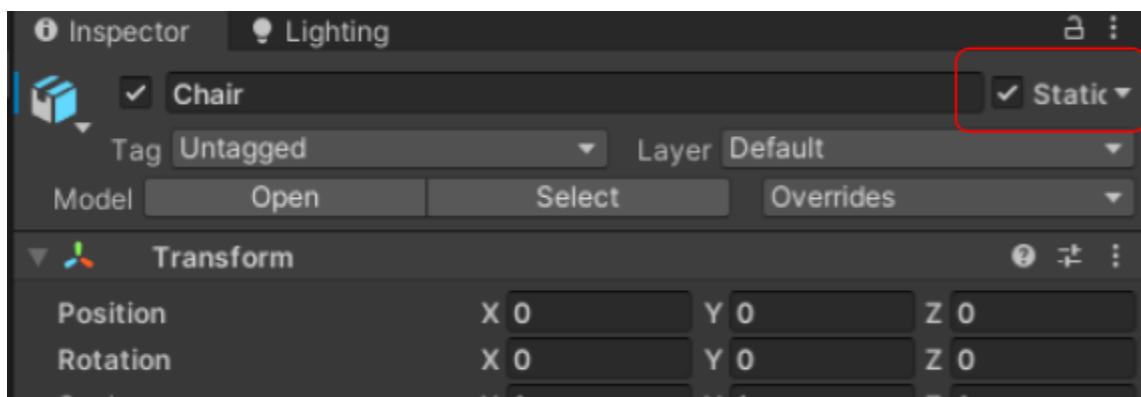
- First you move to the ShadowBake [scene](#) in the [EasyAR->Utilities](#) folder.
- Hope you added your 3d model to the [EasyAR->Meshes](#) folder. For example we take our chair model to generate contact shadow.



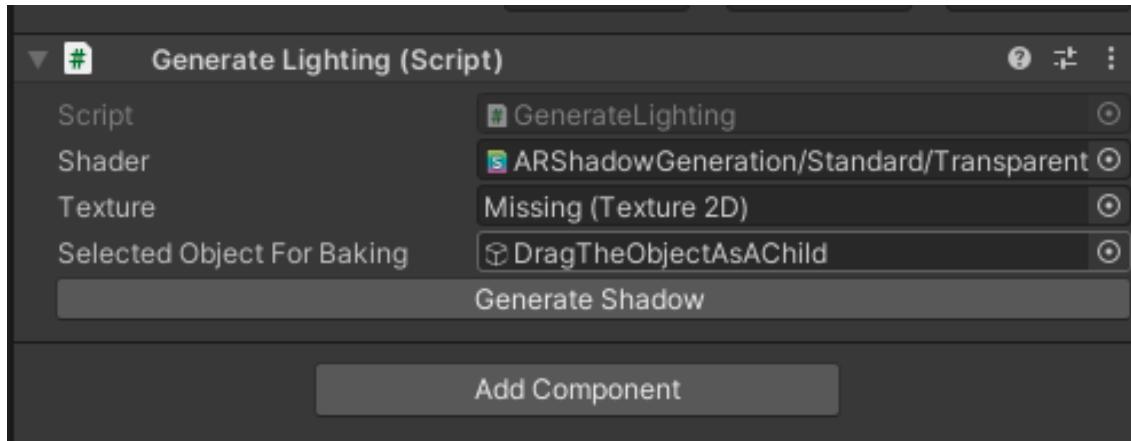
- Drag & drop 3d model (chair) to child of [DragTheObjectAsChild](#) game object in hierarchy.



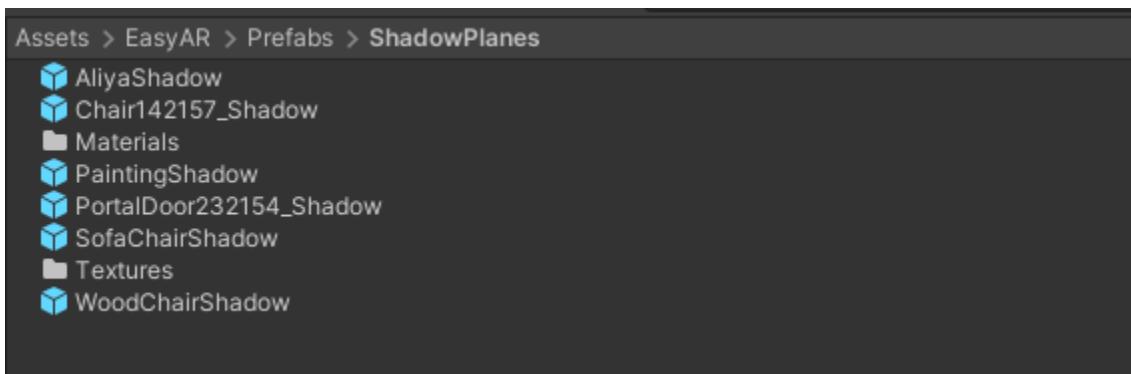
- Then click the 3d model (chair) & go to the inspector. **Check** the static block of the 3d model.



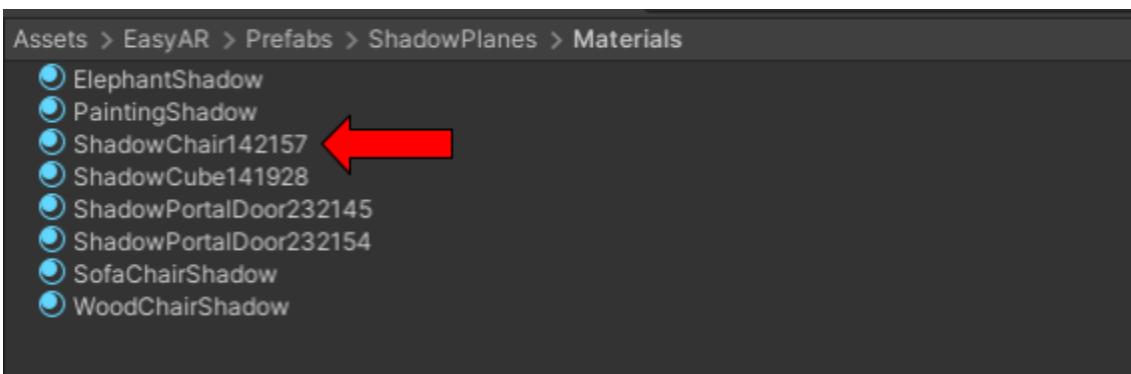
- Select BackShadows gameobject & go to inspector. Under General Lighting Script there is a button called Generate shadow. Click that button.

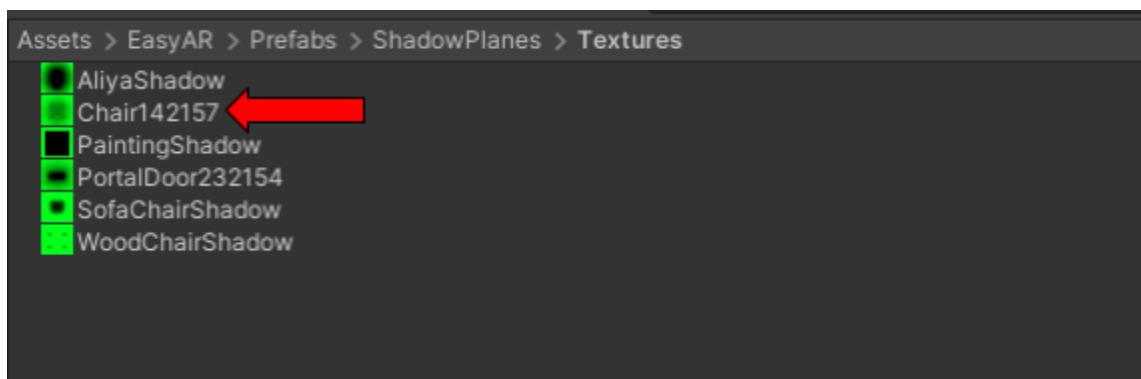


- You can see the process will generate a smooth contact shadow after a few seconds.
- Now you can collect this contact shadow prefab in the [EasyAR->Prefabs->ShadowPlanes](#) folder. Here [Chair142157_Shadow](#) is our newly created shadow map.

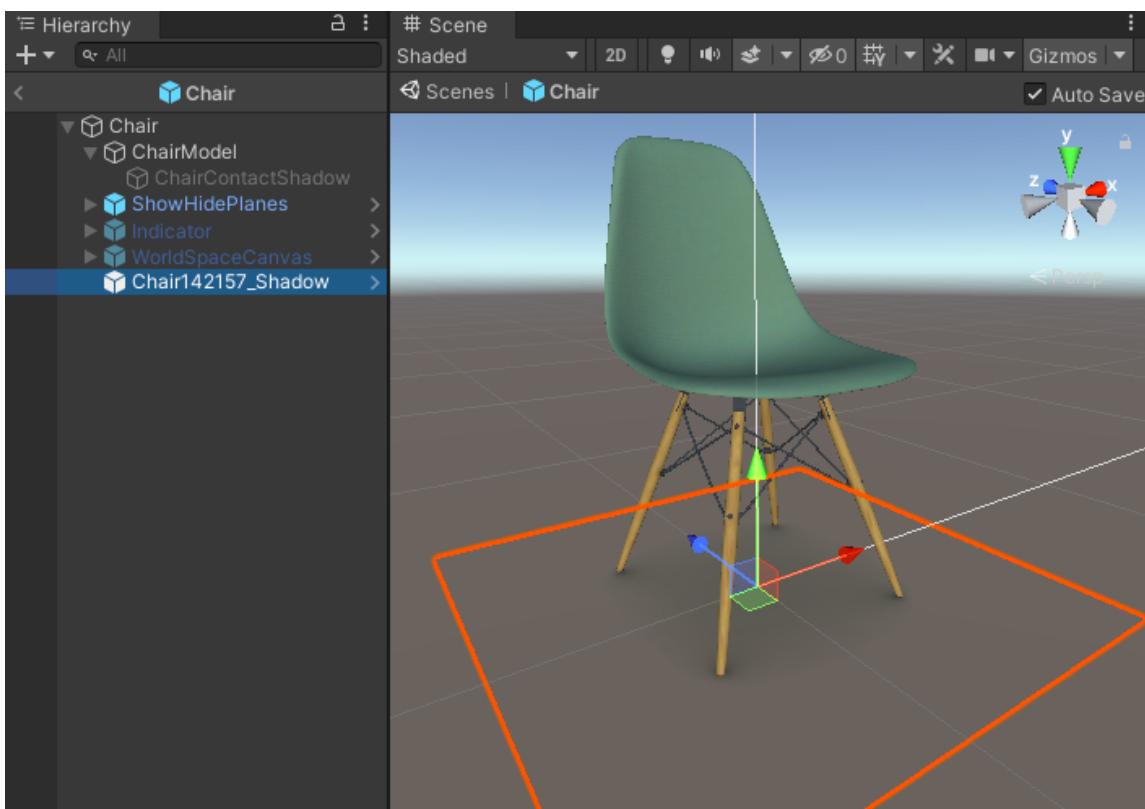


- In the Materials folder you can find that contact shadow's material & Textures folder texture of contact shadow with the same name as prefab saved.





- Now you can add this contact shadow to your gameobject prefab. Select the 3d model prefab (chair prefab) & put it as a child of the 3D model prefab. Then adjust the shadow as you wish.



- Now you can see the smooth contact shadow....

Enjoy the Easy AR.. Thank you!