

Key Concepts

ARCore (Android) and ARKit (iOS) handle many of the tough tasks associated with AR, such as detecting movement and the local environment. AR Foundation is built on top of them, providing a unified interface for AR development.

Plane detection finds horizontal and/or vertical planes in your real world physical surroundings, and creates gameObjects that you can manipulate inside your application.

- Read the docs: [AR Foundation](#), [ARCore](#), [ARKit](#)
- Complete this tutorial on [Plane Detection](#).

[Prefabs](#) allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset.

The **Instantiate function** can be used to create new objects at runtime. Examples include objects used for projectiles, or particle systems for explosion effects.

The [AR Plane Manager](#) is responsible for the detection and creation of plane gameObjects, as well as reporting these events.

[Rigidbody](#)s enable your GameObjects to act under the control of physics. The Rigidbody can receive forces and torque to make your objects move in a realistic way. Any GameObject must contain a Rigidbody to be influenced by gravity, act under added forces via scripting, or interact with other objects through the NVIDIA PhysX physics engine.

[Visual Studio Code for Unity](#)

The [Canvas](#) is the area that all UI elements should be inside and is shown as a rectangle in the Scene View.

The [Image component](#) displays a non-interactive image to the user. You can use this for purposes such as decorations or icons, and you can change the image from a script to reflect changes in other controls.

The [auto layout](#) system provides ways to place elements in nested layout groups such as horizontal groups, vertical groups, or grids. It also allows elements to automatically be sized according to the contained content. For example a button can be dynamically resized to exactly fit its text content plus some padding.

Animation Clips stores keyframe-based animations. Keyframing is the simplest form of animating an object, with it, we only need to know the "key" frames, or conditions, that describe

the transformation of this object, and that all other intermediate positions can be figured out from these.

Animator Controllers are state machines that determine which animations are currently being played and blends between animations seamlessly.

Post-processing is a generic term for a full-screen image processing effect that occurs after the camera draws the scene but before the scene is rendered on the screen. Post-processing can drastically improve the visuals of your product with little setup time.

[Vuforia Dashboard](#)

[Matterport & Vuforia Manual](#)

[Matterport Help](#)

[Area Target Generator in Vuforia](#)

How To:

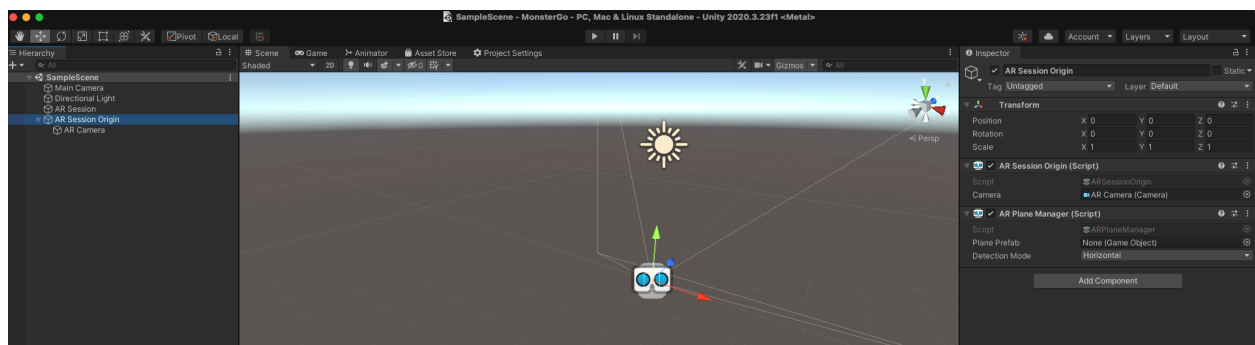
Pokémon Go Application

AR application step up

- Go to package manager and install AR FOUndation and ARCore
- Install AAR Package provided

Setting Up Scene

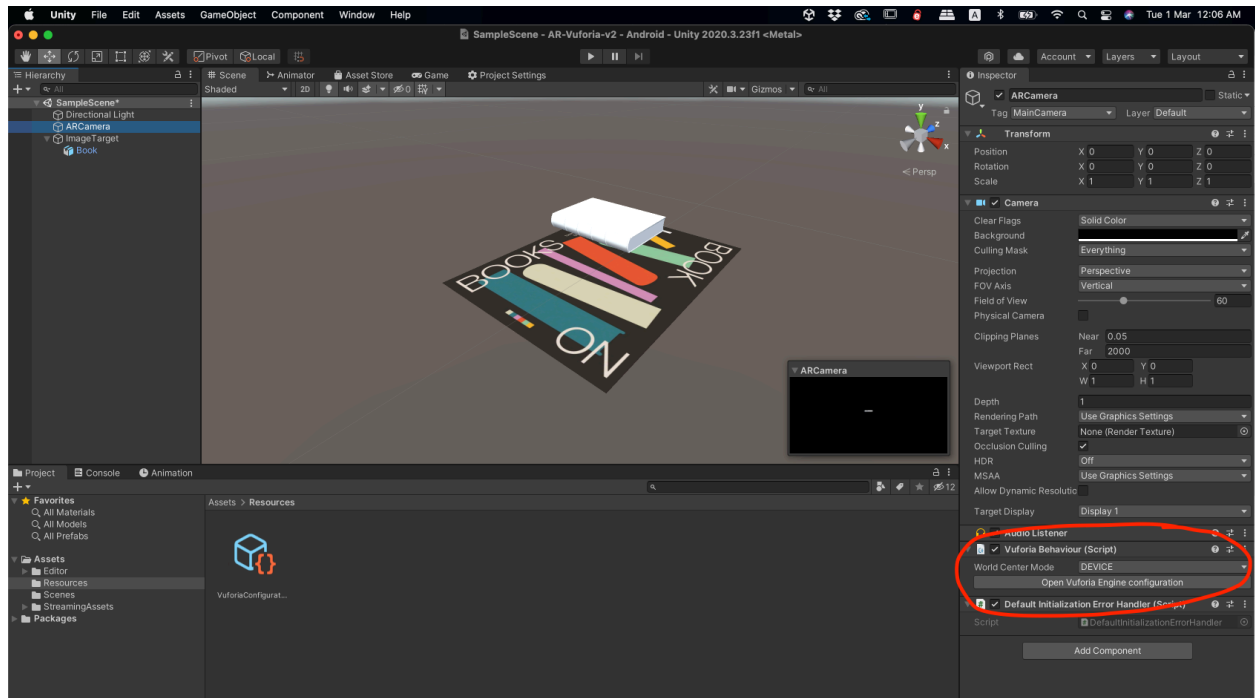
- Go to Hierarchy and right click to XR and set up an AR Session game object. This will configure the AR session you set up.
- Now set XR→AR Session Origin. Right now the Origin is our Camera
- Go to AR Session Origin and in Inspector add AR Plane Manager component.
 - Set Detectio Mode to Horizontal only as we want to add our Pokemons only on Horizontal planes.



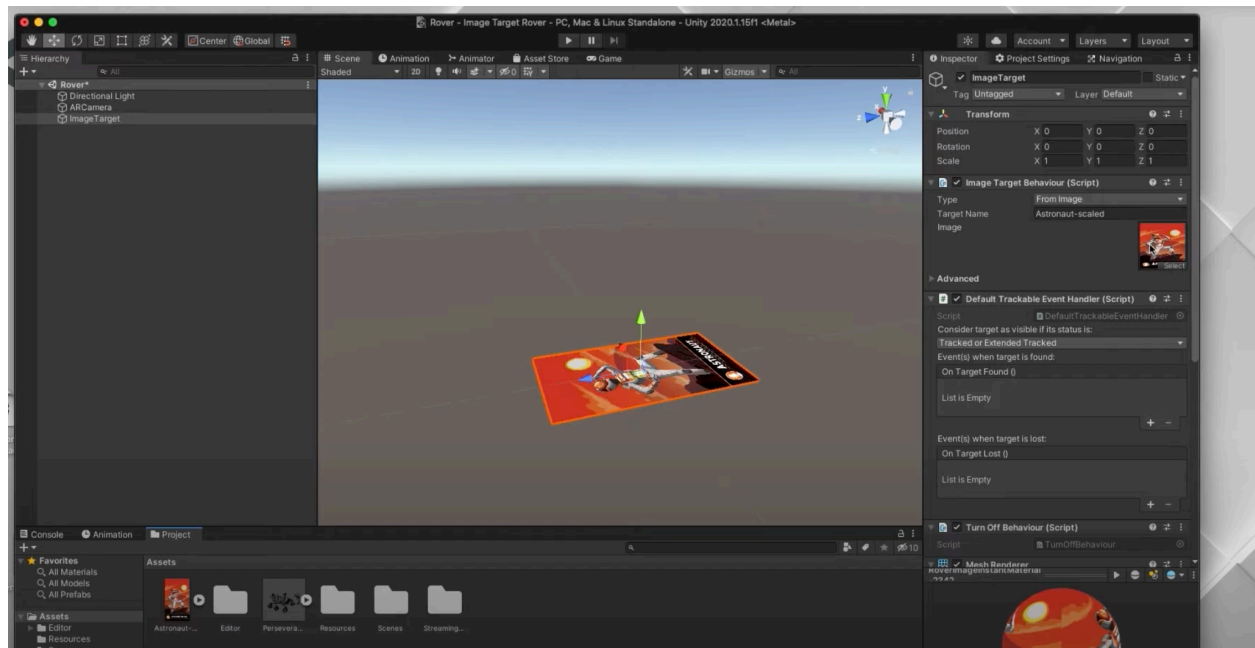
Vuforia

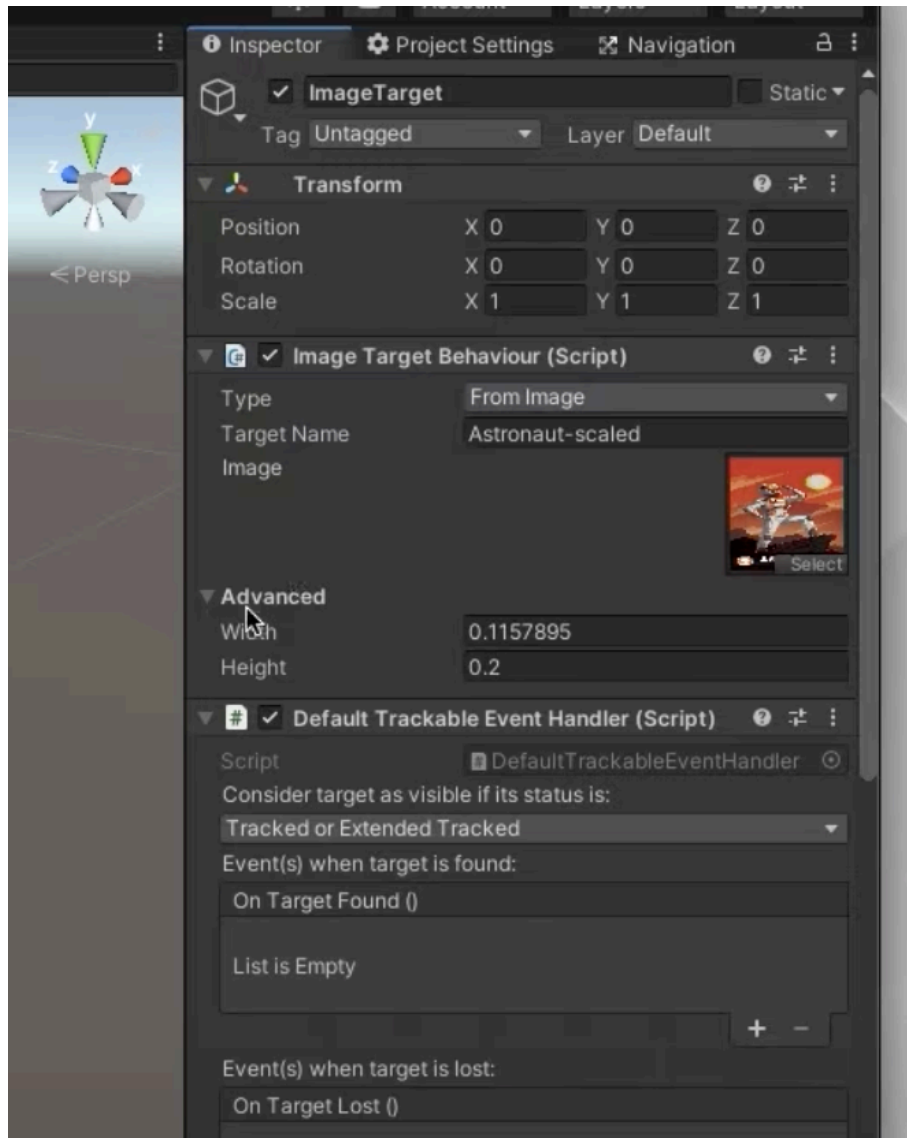
Image Target Single

- Image should be atleast 5inch wide
- Drag Vuforia Unity package to Assets
- In Hierarchy add AR Camera
- Delete Main Camera
- Rename Sample Scene to Main Scene (optional)
- Go to AR Camera→ On Inspector→Vuforia Behavior Script→Open Vuforia Engine configuration
- Add a new License Key from Vuforia developer page for each new project
- METHOD 1
 - Go to Target Manager in Vuforia developer portal
 - Add a database
 - Add an image in the database
 - Add the width of the the image that will be scanned in METERS
 - Download database
 - Drag the database to Assets
 - In Hierarchy→Vuforia→Image Target→Choose database in Inspector
 - Take a FBX Model that you want to augment and drop it into asset folder
 - Drag it as a child of Image Target
 - You can press play button on top of screen and test it
 - To build APK
 - Go to Project settings
 - In Player
 - Change min API Level to Android 10



- Method 2
 - Get an Image Target
 - Go to Inspector:
 - Change Type to FROM IMAGE
 - Set Target Name
 - Drag an image to Assets
 - Go to it's Inspector Panel
 - Change Texture Type: Sprite(2D&UI)
 - Hit Apply
 - Select Image Target
 - Drag image from Assets to Image in the Inspector panel of ImageTarget
 - Under Advanced tab set Width in meters
 - Like before drage Augmented Image as child





Multiple Image Target

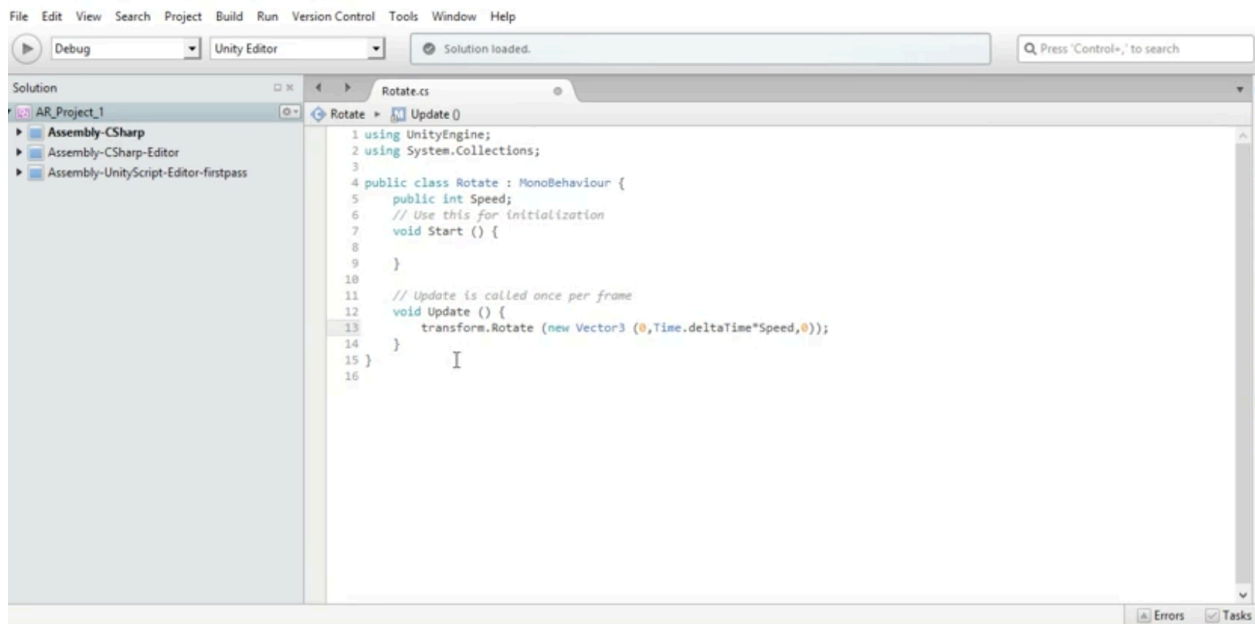
- After creating one image
- Go to AR Camera → On Inspector → Vuforia Behavior Script → Open Vuforia Engine configuration
- Change Max Simultaneous Tracked Images & Objects to 3
- In Vuforia Developer Target make a database with more than one target.
- Download database
- Import it in Unity
- Create Multiple Image Targets
- Use the same process

Area Targets

- [Tutorial 1](#) - using Vuforia Area Target Creator IOS app
- [Tutorial 2](#) - Using Matterport

Rotate Target

- Drag this script on the AR object you want to rotate.

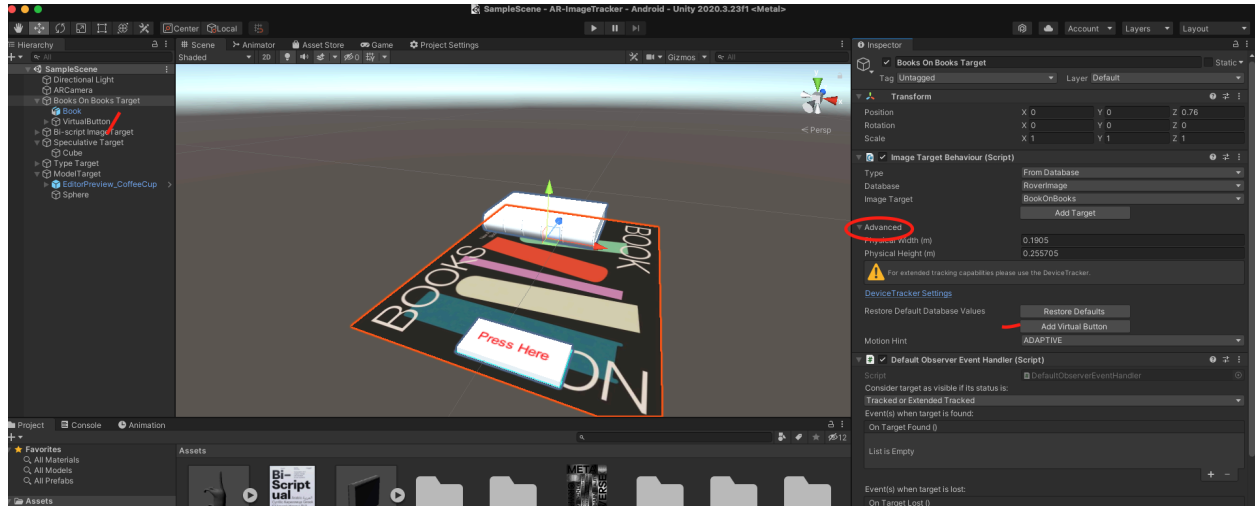


Video Target

- Make a cube as parent of the target image/object
- Transform the cube in the size of a screen with thin z axis
- Drag the video on to assets
- Drag the video to the cube
- Change properties on inspector panel

AR Button

- Only works for image/object targets that are done through METHOD 1 through Vuforia website
- Go to the Image Target. Then go to its Inspector panel.
- Press Add Virtual Button in the Advanced dropdown.
- Place the button at the right place
- Go to the image target and add a script to it. (Script in file)
- Now add a cube & text to the child of the button to show something visually.



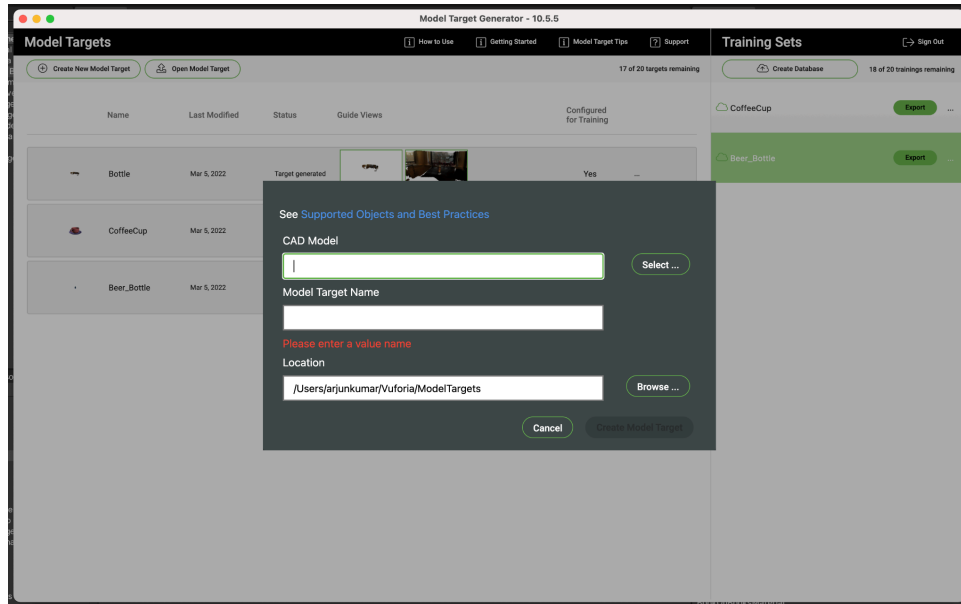
Model Target

<https://developer.vuforia.com/downloads/tool>

<https://library.vuforia.com/model-targets/introduction-model-targets-unity>

<https://library.vuforia.com/model-targets/model-target-guide-view>

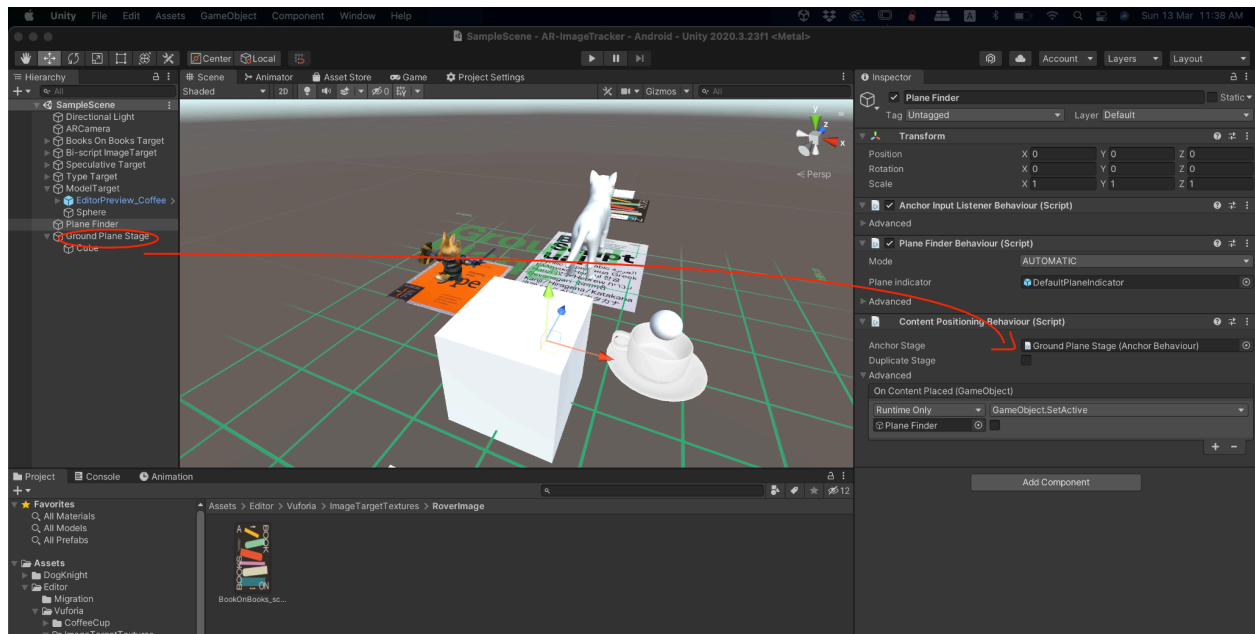
- Download Model Target Generator App on MacOS
- Download a model from turbosquid or sketchfab that you want to target
- Create a New Model Target that can be read by Vuforia



- Once the model has been created a Unity file will be created for that object.
- Drag that unity file into Unity
- In Hierarchy create a new model target within the Vuforia AR Engine
-

Ground Plane Detection

- Right click on Hierarchy and add Vuforia Plane Finder. Vuforia Engine → Ground Plane→Plane Finder
- Then add the Ground Plane Stage too. Vuforia Engine → Ground Plane→Ground Plane Stage
- Add the object that you want to augment on the ground plane as a child of the Ground Plane Stage.
- Go to Plane finder. In Inspector, add the ground plane stage to the anchor stage.



- Open Advanced and add the Plane finder as the Runtime Only object.
- Set function to `GameObject.SetActive`. This is to prevent object change location on touch.

ARCore