AR Reference Document

I. SETUP

❖ Requirement

- Unity **2019.4.x** (LTS) (x is from 11 or more)
- Use **Unity Standard Render Pipeline** (use **Universal Render Pipeline** for bonus points)
- Unity Package Manager's **AR** packages
- **Android** platform (iOS for bonus points)
- Free game model and animation can be downloaded at https://www.mixamo.com/
- Free environment can be downloaded at https://assetstore.unity.com/packages/3d/environments/free-low-poly-desert-pack-106709

❖ Preparation

- Download Unity's **AR Foundation** sample project **version 3.1** from https://github.com/Unity-Technologies/arfoundation-samples/tree/3.1
- Extract and open the sample project with Unity 2019.4.11, Android Platform.
- Resolve built-in packages errors if encountered.
  - Upgrade **TextMeshPro** Package to verified version
  - Remove **Unity Collaborate** package
  - If the errors still remain: remove any other packages causing errors on Console window
- Make sure the required packages for AR are installed - PackageManager:
  - AR Foundation 3.1.3
  - AR Subsystems 3.1.3
  - ARCore XR Plugin 3.1.3
  - ARKit XR Plugin 3.1.3
  - XR Plugin Management 3.2.15
- Go to Project Settings -> XR
  - Create both **ARCore** and **ARKit** Build Settings files.
- Set both's Requirement to **Optional**

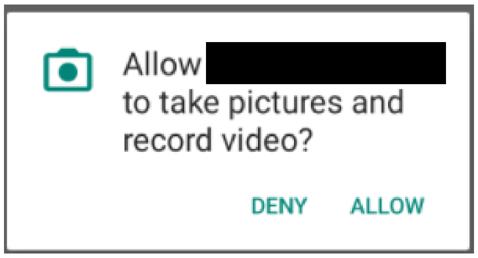
II. REQUIREMENTS

❖ Scenes

The game now should have 2 scenes:

- **Main** scene of the original game
- **AR** scene of the game with AR mode

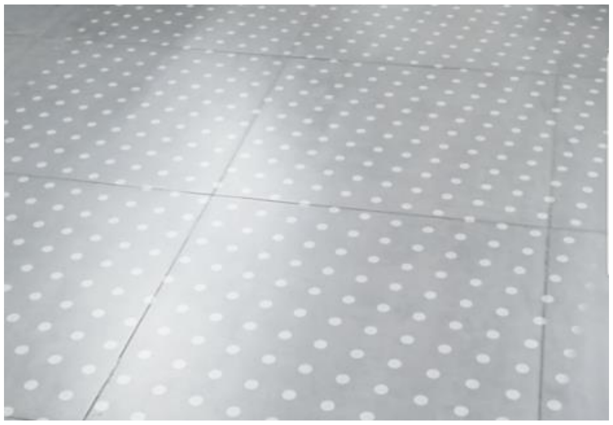
- **Main Scene**:
  Should have UI button to triggers **Android Native Camera Permission** popup when have not permission yet:
  - If press **Allow**, then load the AR Scene
  - If press **Deny**, then keep playing in the Main Scene



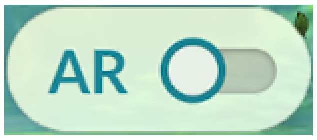
Allow [redacted] to take pictures and record video?

- **AR Scene**:
  The AR Scene should be able to:
  - **Detect** real world horizontal surfaces
  - **Place** the ball game on real world surfaces
  - Allow the user to **interact** with the game normally

➢ *Requirements (must have):*
  - Has UI button to toggle the **AR Plane Visualizer**



  - Has UI button to turn on/off the **real world environment render** from the device's camera.



  - The game after rendered on a surface must stick at there without changing position on new detected plane or else.

➢ *Bonus points (nice to have):*
  - When the user uses 2 finders to do the **pinch zoom**
    - The game board should be scaled up or down following the user fingers distance.

- When the user **swipes horizontally**
  - The game board should be rotated to the left or right (Y axis) depending on the swipe direction.
- **Shadow** for AR Objects:
  - Create a material for the AR Plane that is completely transparent but can still receive shadow for the game board.

(Just an example image about AR Shadow)

- **Universal Render Pipeline:**
  Upgrade the whole project to **Universal Render Pipeline**
  - Make sure the AR Camera still works well with new pipeline
  - Make sure humanoid and environment materials still work well
  - Make sure the Transparent Plane Material above still works well

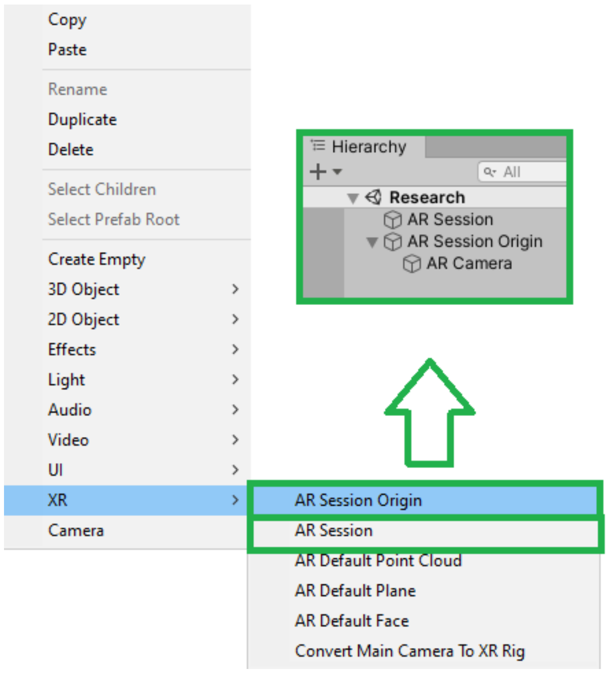
III. INSTRUCTION

❖ In the Scene Hierarchy, create **AR Session** and **AR Session Origin**

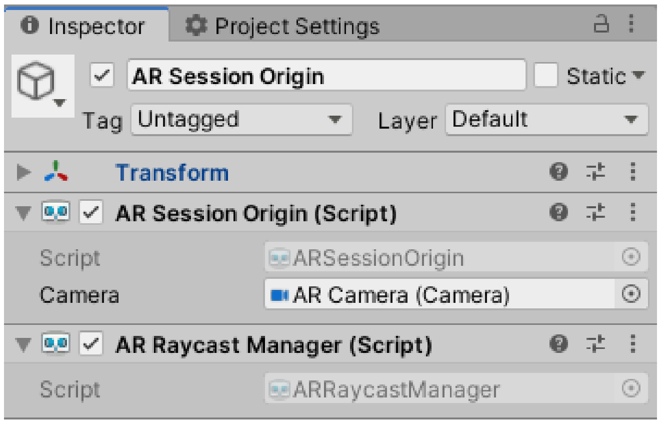
- ***AR Session*** controls the lifecycle of an AR experience, enabling or disabling AR on the target platform

- ***AR Session Origin*** mission is to transform the trackable object from the real world - called Session space provided by an AR device, into the Unity scene World space with correct place (position, rotation and scale attributes)

- ***AR Camera*** used to render any trackable objects we need to visualize on the device screen. It has a component to enable the real world background.



- Then add the **AR Raycast Manager** component into the **AR Session Origin** which will allow us to do **Raycast** and interact with the Trackable Object (In this practice is the detected **PlaneWithinPolygon**).

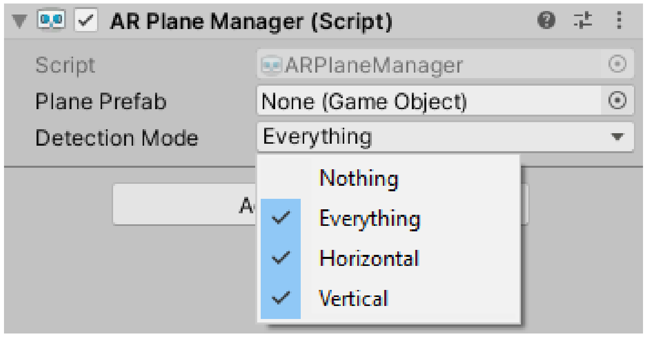


AR Session Origin
Static
Tag Untagged
Layer Default

Script ARSessionOrigin
Camera AR Camera (Camera)

Script ARRaycastManager

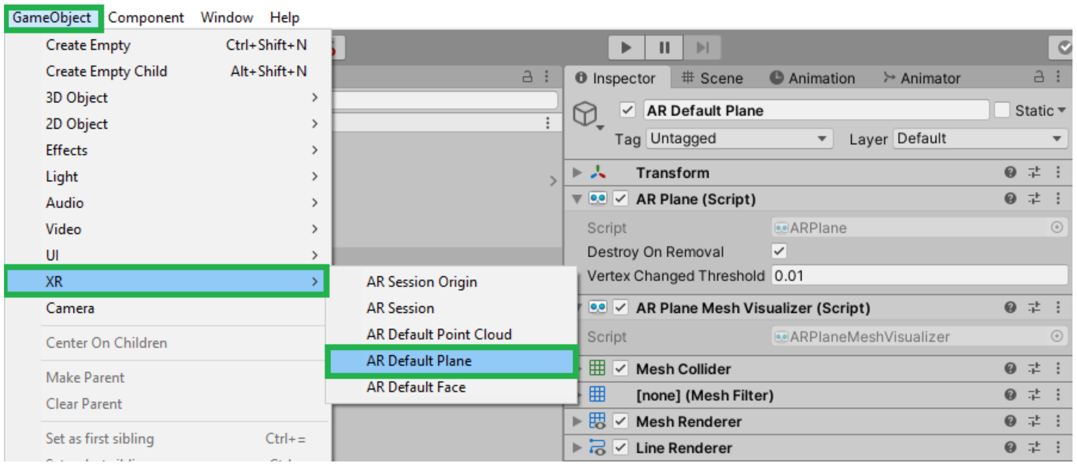
- To have the Plane detection, add the **AR Plane Manager** into **AR Session Origin**



Script ARPlaneManager
Plane Prefab None (Game Object)
Detection Mode Everything

  - When a surface is detected, the **planesChanged** event of an **ARPlaneManager** component will be raised
  - Also the **boundaryChanged** will raise when the detected plane got updated.

- Finally, create the **Plane Prefab** for the **AR Plane Manager** above from the default XR Object in the menu.



AR Default Plane
Static
Tag Untagged
Layer Default

Script ARPlane
Destroy On Removal ✓
Vertex Changed Threshold 0.01

Script ARPlaneMeshVisualizer