AR Shadow — Unity Asset by makaka.org

AR Shadow (Unity Asset) implements simple real-time shadows for apps with Augmented Reality. This is Unity shader for transparent surfaces.

Demo Project included: real Plane Detection (AR Foundation Engine: ARKit, ARCore) and Object Placing. **AR Safety First.**

> *Demo Template is an Excellent Starting Point to Create Apps and Games with Augmented Reality.*

Contents

Package Contains

- ⭐ Complete Demo Scene with Tutorial, real Plane Detection, visual Plane Confirmation/Resetting, and Object Placing in a selected point on a detected plane.
- ⭐ Menu Scene with AR Safety Tutorial.
- ⭐ Permission Scene with Camera Permission Request using free Native Camera.
- ⭐ Loading Screen to switch scenes seamlessly.

Features of AR Shadow & Plane Detection Demo Project

Bring the enchanting Power of Augmented Reality into your amazing AR Game or App:

- ⭐ AR Onboarding UX with Transparent Video Manuals & AR States;
- ⭐ AR Light Estimation: estimates light data in physical space and applies it to game space;
- ⭐ AR Safe Zone. The player needs to stay in the Safe Zone to avoid accidents during the game and continue the game itself! The player will be notified when leaving the Safe Zone;
- ⭐ AR Foundation Support Checker allows assigning your own experience when AR is not supported (by default, it shows informational message and link to requirements);
- ⭐ Editor Testing: “Spectator Mode” allows simulation of smartphone motion (translational (WASDQE keys) & rotational (Right Mouse Button));
- ⭐ Mobile Optimizations:
  - ⭐ TextMesh PRO for Texts to update them when really needed.

AR Shadow Shader is a part of Unity Assets

AR Survival Shooter (docs).

VR + AR: Mixed Reality (MR) (docs).

AR Toss Boss (docs).

AR Basketball GO (docs).

AR Throwing (docs).

AR Camera Lite (docs).

Tutorial

> *This tutorial is relevant for 3.0+ version of Asset.*
> *Tutorials for previous versions can be found in the asset folder.*

Getting started with AR Shadow

Folders & Files in the package by default:

- Makaka Games,
- XR.

Steps

> *If you have any issues with the first launch then just Reach Support with Invoice Number and Get Help.*
>
> *If you read this tutorial from PDF, first check the latest docs online to get actual information.*

1. Create a new Unity Project with Unity 2021.1.20.
2. File > Build Settings > iOS/Android > Switch Platform.
3. Next Packages must be manually installed with Git & Unity Package Manager:
    1. Native Camera 1.3.0.
4. Download and import AR Shadow into Unity.
    1. Warning Windows:
        1. Click “Import” to overwrite the Project Settings with predefined ones.
        2. Click “Install/Upgrade” for Package Manager Dependencies.
        3. Click “No” when enabling the backends for the new input system.

5. Next Packages are provided with Unity Package Manager, and they are already installed for this Asset by default. If packages are missing (Warning Window did not appear) then install them again with Unity Package Manager (with advanced settings enabled: “Pre-release Packages” & “Show Dependencies”):

    1. TextMesh Pro 3.0.6:

        1. **Always Required**: Window > TextMeshPro > Import TMP Essential Resources.

    2. AR Foundation 4.1.7;
    3. AR Subsystems 4.1.7;
    4. ARCore XR Plugin 4.1.7;
    5. ARKit XR Plugin 4.1.7;
    6. Editor Coroutines 1.0.0;
    7. Input System 1.0.2;
    8. Subsystem Registration 1.1.0;
    9. XR Legacy Input Helpers 2.1.8;
    10. XR Plugin Management 4.1.0.

6. Reopen Unity Project.
7. Open Scene: Makaka Games > AR > AR Shadow > Scenes > Demo.
8. Test in the Unity Editor or Build for Mobile.

Object Placing in AR

Simple Cube is the Object for Placing in AR by default.

To set your own Game Object in Unity Editor, place it instead of this Cube in the same position on *ARGround* game object, keeping the hierarchy because the project is sensitive to it.

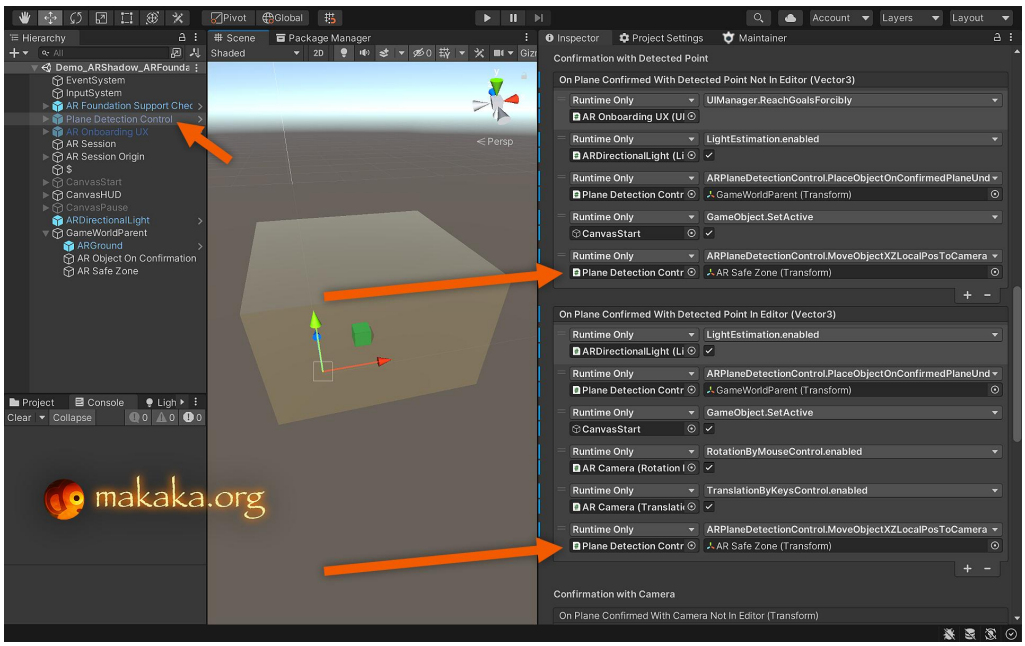
AR Safe Zone

The player needs to stay in the Safe Zone to avoid accidents during the game and continue the game itself! The player will be notified when leaving the Safe Zone.

Position

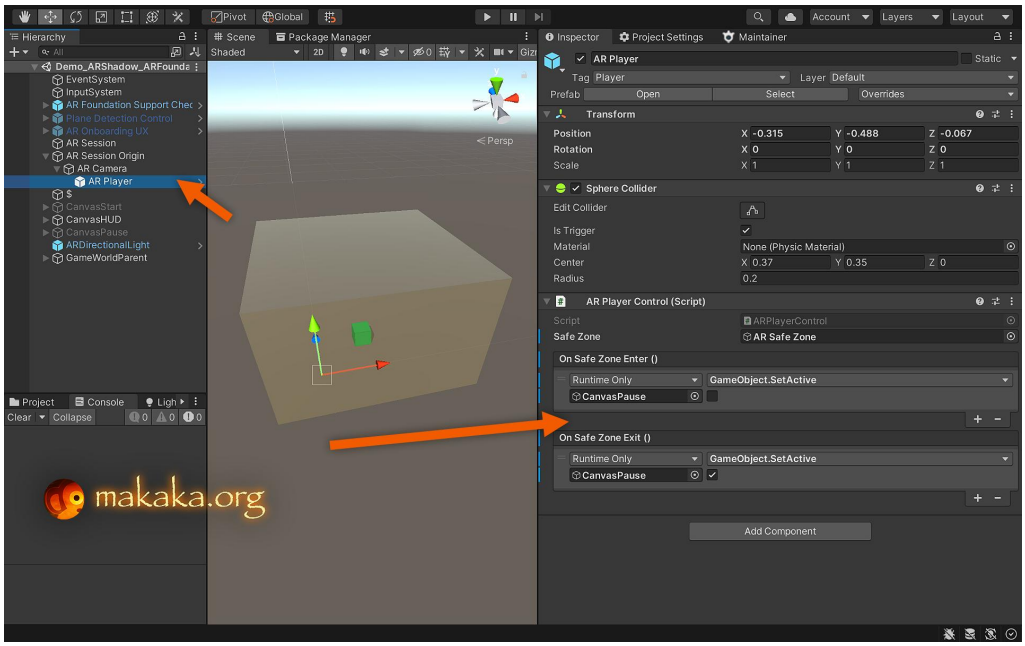
By default, the Safe Zone is placed around the Player (AR Camera) on *ARGround* game object with a real approximate height of 2 meters and a real approximate width of 3.5 meters. It's enough to place and observe your AR Object from all sides. You can change the size of the Safe Zone by changing its scale.

Every AR Experience is Unique, and you can also place the Safe Zone around the Placed Object: just delete the function call which moves the Safe Zone to the Camera position: *Hierarchy View > Plane Detection Control: Confirmation with Detected Point (section)* > Both Events: for Editor and Not.



Events

Reactions to Entering and Exiting the Safe Zone can be assigned in *Hierarchy View > AR Session Origin > AR Camera > AR Player: AR Player Control*.



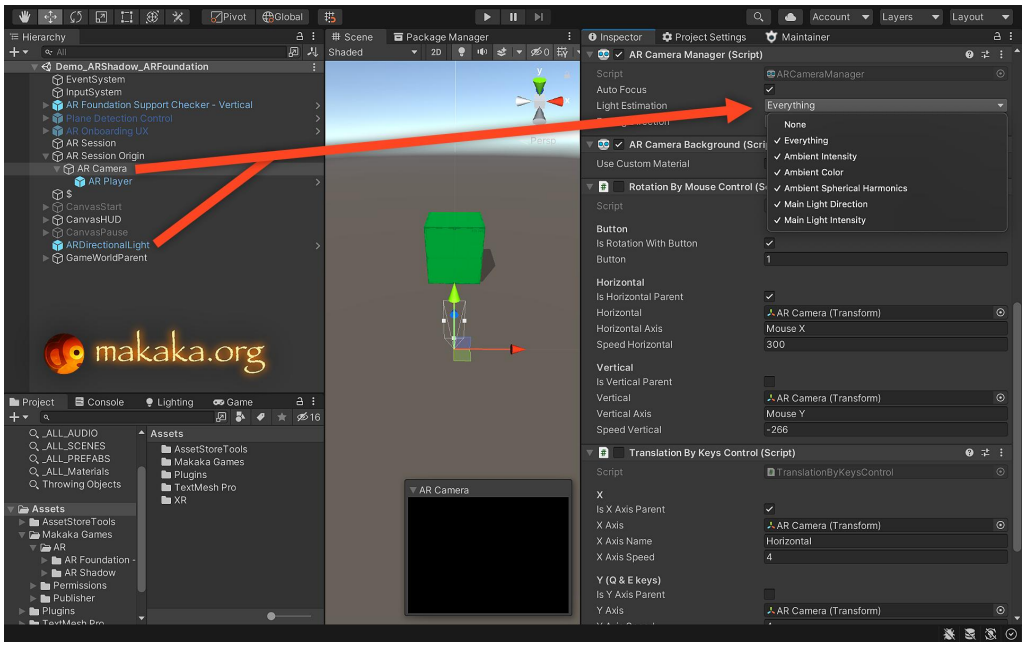
AR Light Estimation

The mechanism estimates light data in physical space and applies it to game space **depending on what is supported by a particular smartphone**. All options are enabled by default:

- Ambient Intensity,
- Ambient Color,
- Ambient Spherical Harmonics,
- Main Light Direction,
- Main Light Intensity.

On iOS, when AR Mode is not Face Tracking, then only Ambient Intensity and Ambient Color are supported.

You can turn it off in *Hierarchy View > AR Session Origin > AR Camera: AR Camera Manager > Light Estimation*. Control Script is attached to the Light game object.



AR Shadow and Any AR Engine

AR Shadow works independently and does not require a specific AR engine. E.g., you can transform your game scene into Augmented Reality in 4 minutes with AR Shadow and AR Camera Lite (docs).

Testing

Unity Project provides Basic Editor Testing (without smartphone & plane detection) to imitate smartphone **motion** (such way you can test AR Safe Zone & observe AR Object from all sides):

- **translational**: WASDQE keys;
- **rotational**: Right Mouse Button.

> *Use Fullscreen of Game View in Unity Editor while testing to get a seamless experience.*

You can forcibly test the case when AR is Not Supported by checking the next flags in *Hierarchy View > AR Foundation Support Checker*:

- *Is Checked In Editor On Init*;
- *Is AR Unsupported Not In Editor Test*.

Tested with Mobile Devices

- ⭐ iOS on iPhone XS Max.
- ⭐ Android on Samsung Galaxy A71.

Support

First, read the latest docs online.
If it didn't help, get the support.

Changelog

Check the current version on Asset Store.
The latest versions will be added as soon as possible.

3.0 (Complete Template — Excellent Starting Point to Create AR Apps and Games)

Features:

- ⭐ AR Onboarding UX with Transparent Video Manuals & AR States;
- ⭐ AR Light Estimation: estimates light data in physical space and applies it to game space;
- ⭐ AR Safe Zone. The player needs to stay in the Safe Zone to avoid accidents during the game and continue the game itself! The player will be notified when leaving the Safe Zone;
- ⭐ AR Foundation Support Checker allows assigning your own experience when AR is not supported (by default, it shows informational message and link to requirements);
- ⭐ Editor Testing: "Spectator Mode" allows simulation of smartphone motion (translational (WASDQE keys) & rotational (Right Mouse Button));
- ⭐ Menu Scene with AR Safety Tutorial;
- ⭐ Start Tutorial after Plane Confirmation;
- ⭐ Using of Native Camera to check and request Camera Permission before the Demo Scene.

Improvements:

- Unity 2021.1.20;
- Plane Detection Tutorial: Loading Animation, Clearer Guidance;
- Modern Setting Standards of Project Settings (based on New Unity Project).

2.1:

- Improved Demo with tutorial, visual plane confirmation/resetting & placing the Game Object (Cube);
- Restart Button;
- Unity 2021.1.12;
- TextMesh Pro for All Texts.

2.0:

- Unity 2021.1.