
unity
Unite
Seoul
2019

Using AR Foundation for Multi-Platform AR

Ashley Alicea
Evangelist, Media & Entertainment – Unity Technologies
ashley@unity3d.com
@avashly

unity
Unite Seoul 2019

Overview

- Defining Handheld AR
- Core Features of ARCore & ARKit
- AR Foundation Features
- Recent AR Foundation Updates
- Getting Started with AR Foundation
- Live Demo & Resources

Handheld AR

Augmented reality powered by phones & tablets

Core Features of Mobile AR SDKs

Plane Finding

Using the sparse point cloud extraction from the SDKs to estimate and create planes

Position Tracking

Tracking the device’s position as it moves throughout the space

Light Estimation

Estimate the current camera views ambient light value to light digital objects with real world light

ARCore Unique Features

- Oriented Feature points (direction vectors for feature points)
- Cloud Anchors
- Face Tracking (computer vision)
- Instant preview
- Augmented Images

ARKit Unique Features

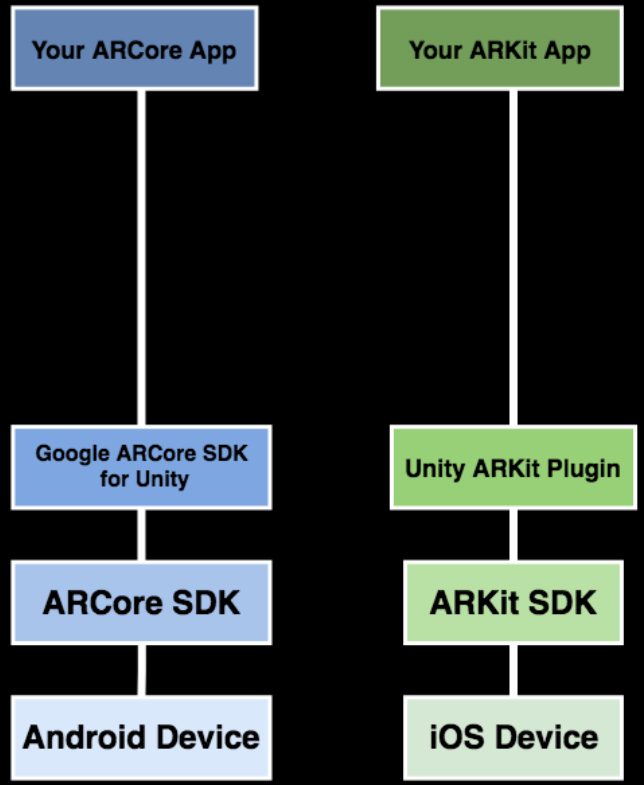
- World Maps
- Trackable Images
- Object scanning & recognition
- Face tracking (blend shapes)
- Environment Probes


AR Foundation
For ARCore & ARKit

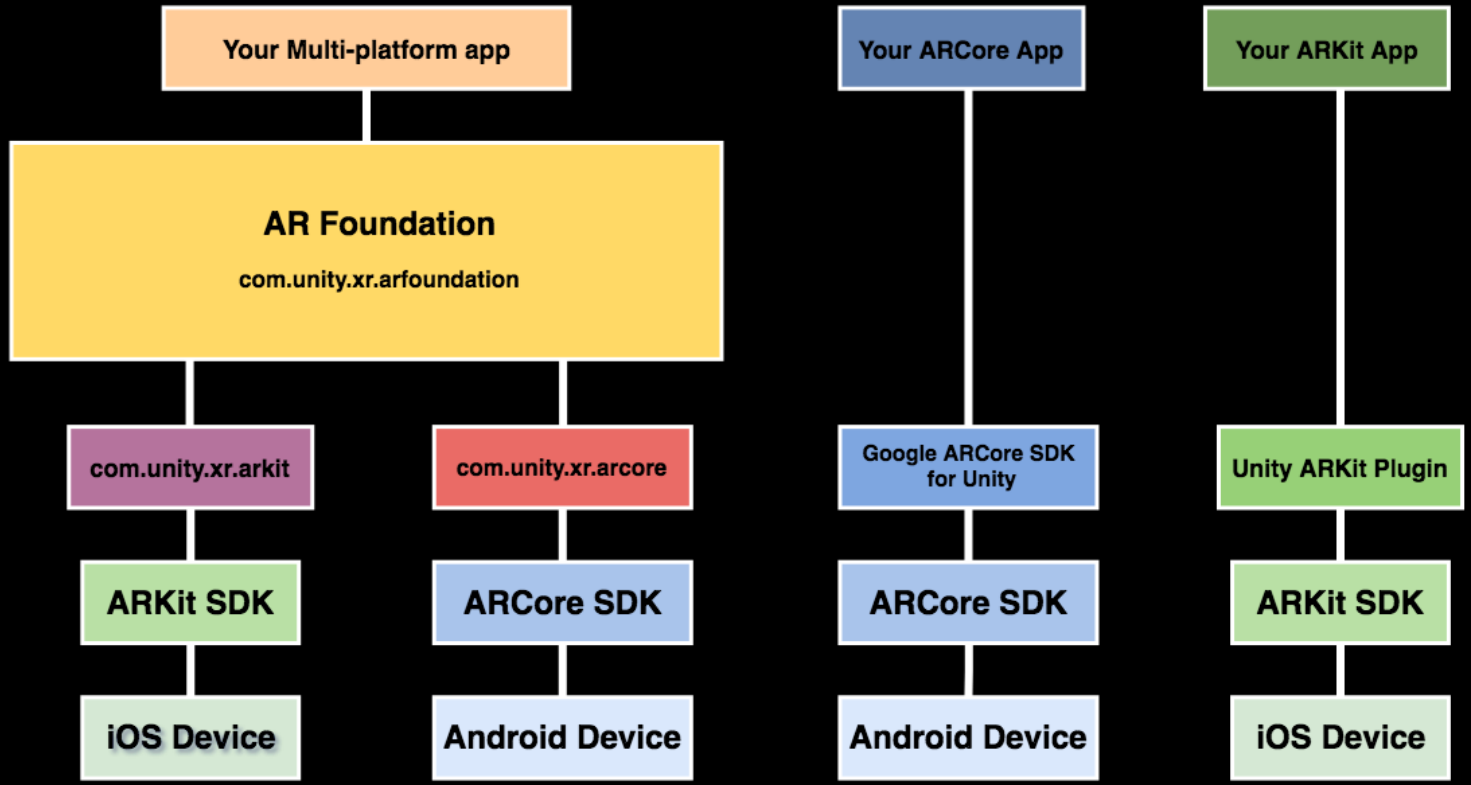
What is AR Foundation?

- Multi-platform API for ARCore & ARKit apps
- Uses core functionality shared between both platforms
- Develop your app once & deploy to both devices

AR Foundation



AR Foundation



AR Foundation: Today

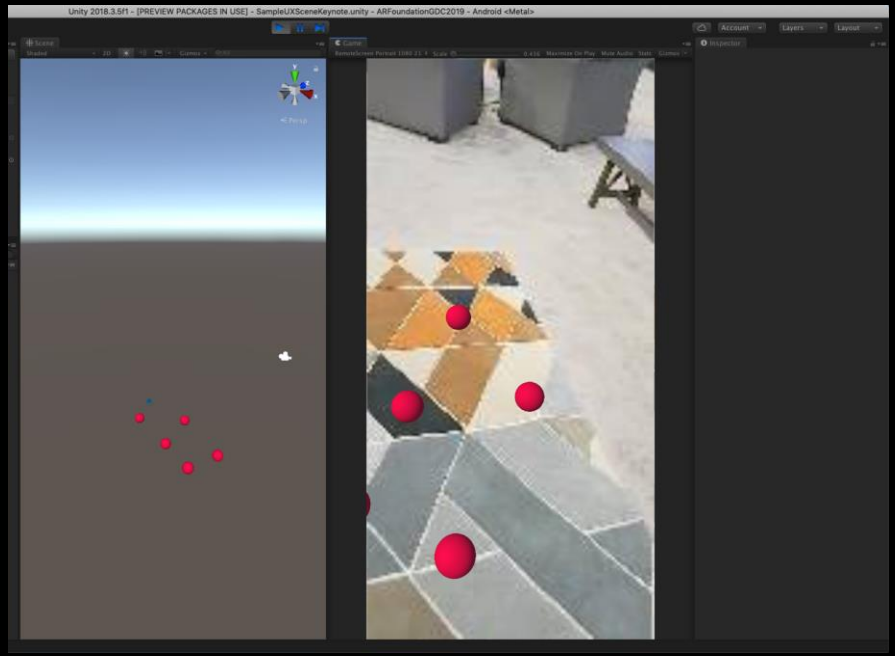
- Preview package
- Supports core functionalities of ARCore / ARKit
- Camera Image APIs
- Preview support for LWRP 4.8.0
- AR Foundation Samples project available to reference

AR Foundation: Today

Supported Features	ARFoundation	Google ARCore SDK for Unity	Unity ARKit Plugin
Plane Detection (Horizontal and Vertical)	✔	✔	✔
Feature Point Detection	✔	✔ + Oriented Feature Points	✔
Light Estimation	✔	✔ + Color Correction	✔ + Color Tempature
Hit Testing (Feature point and Plane raycasting)	✔	✔	✔
AR Anchors	✔	✔	✔
Image Tracking	coming soon	✔	✔ + Image Tracking
3D Object Recognition	coming soon		✔
Enviroment Probes	coming soon		✔
World Maps	✔		✔
Face Tracking (Anchor, Mesh, Blendshapes)	✔ - ARKit	✔	✔ iPhone X + Variants Only
Cloud Anchors	coming soon	✔	
Remoting	in preview	✔ - Instant Preview	✔ - ARKit Remote
Simulation	in preview		
LWRP support (+ Shader Graph*)	4.8.0 supported	coming soon	coming soon
Camera Image API	✔		

AR Foundation: Tomorrow

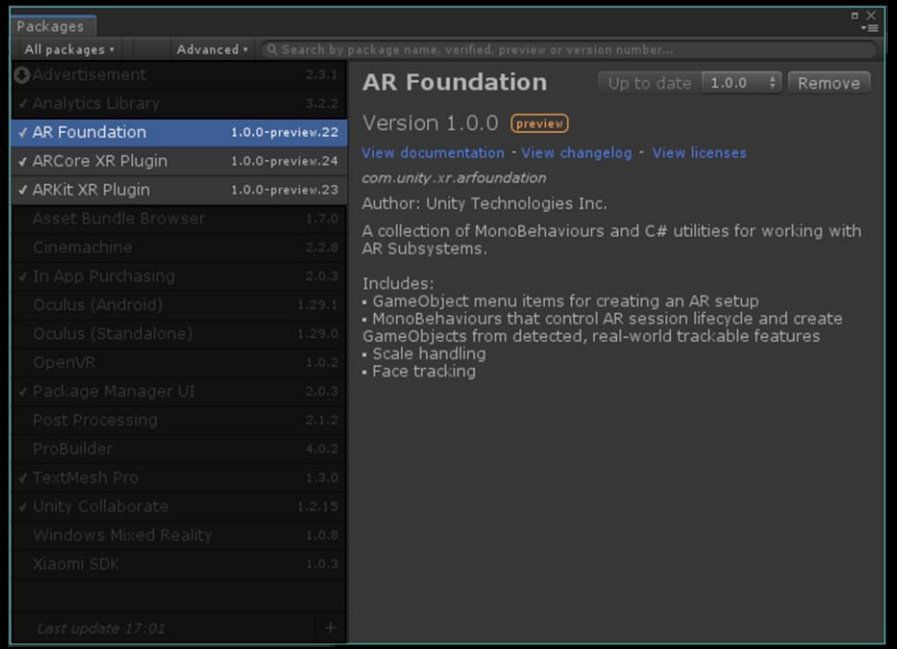
- Feature parity with ARCore 1.7 & ARKit 2.0
- Wearable support
- AR Remote (with session recording & playback)
- Simulation



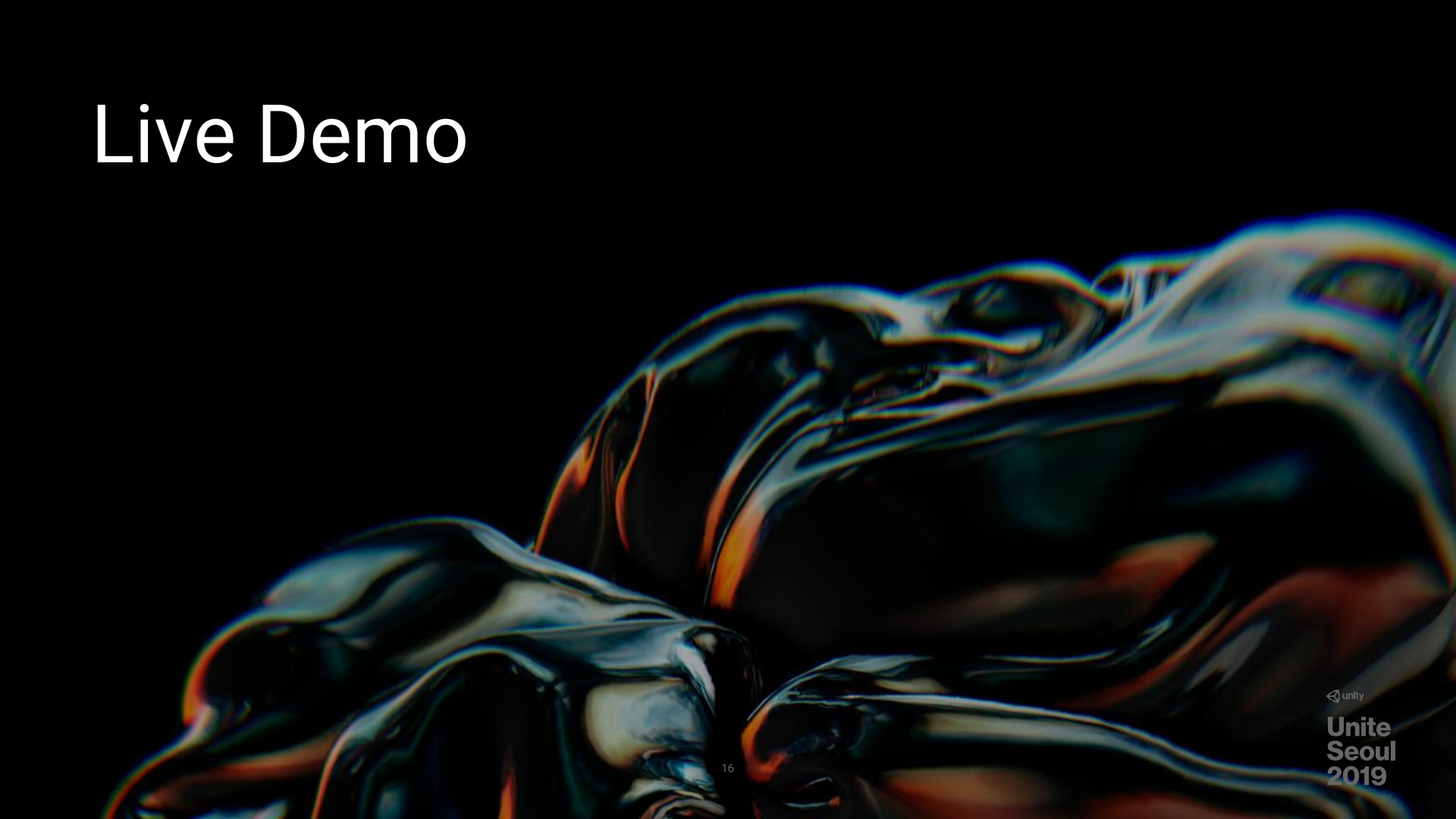
Where to Get AR Foundation

Available now in Unity 2018.1+
via the Package Manager!
Be sure to download:

- AR Foundation
- ARCore XR Plugin
- ARKit XR Plugin



Live Demo



Shader Graph
‘Toon’ Light Estimation

- PBR Shader Graph auto receives Light Data
  - Light Data:
    - Directional Light
    - Additional Lights
    - Ambient Light
    - Shadows
  - Light Estimation applies to Light Data
    - Directional Light Color & Intensity

Shader Graph ‘Toon’ Light Estimation

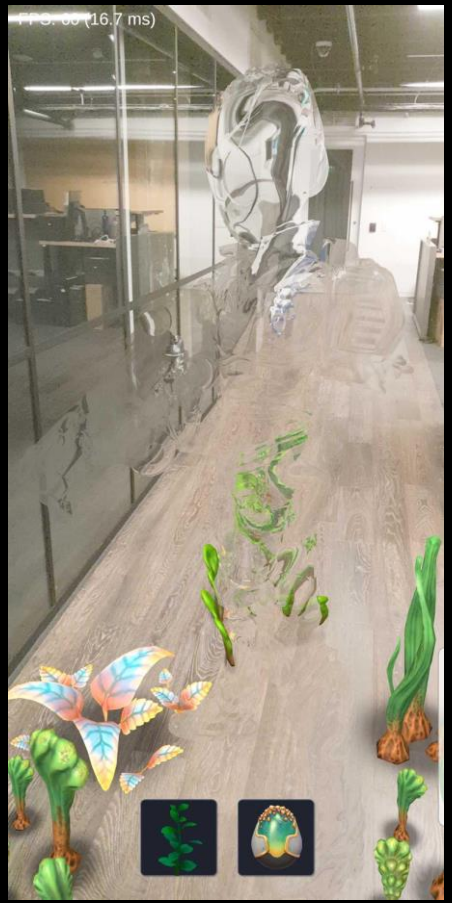
- Default Result: Too Dark
- Shader Graph Additions:
  - Custom Light Nodes
  - Combine Light Node Data with Albedo Map
  - Multiply Influence Amount by 0.5 (Shadows & Light Falloff still visible)
  - Output through Emission
  - github.com/PauloPatez/ShaderGraph_LightNodes

No Light Estimation

With Light Estimation

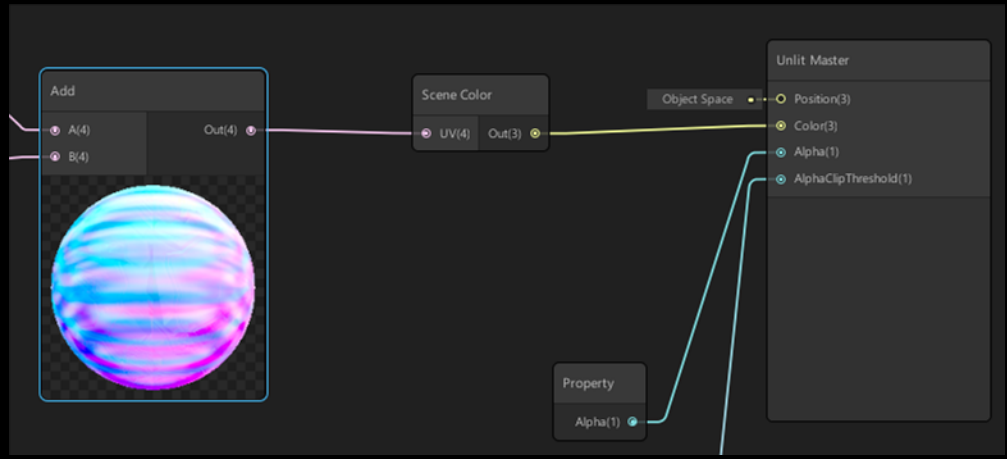
Shader Graph
AR Refraction

- LWRP Asset: Opaque Texture
  - Texture of everything drawn in Opaque Pass
  - Used for GrabPass-like Effects
- AR Refraction
  - Camera Feed Texture and Opaque Objects are both included in Opaque Texture
  - Can use Opaque Texture for AR Camera Effects



Shader Graph Refraction

- Scene Color Node
  - If Transparent Shader, accesses Opaque Texture
  - Pass in Screen Position UV
  - Output as RGB Color
- Terraformer's Refraction
  - Combine Scrolling UV Texture with Terraformer's Normal Map
  - Apply to Opaque Texture, using Scene Color Node



AR Shadow Plane

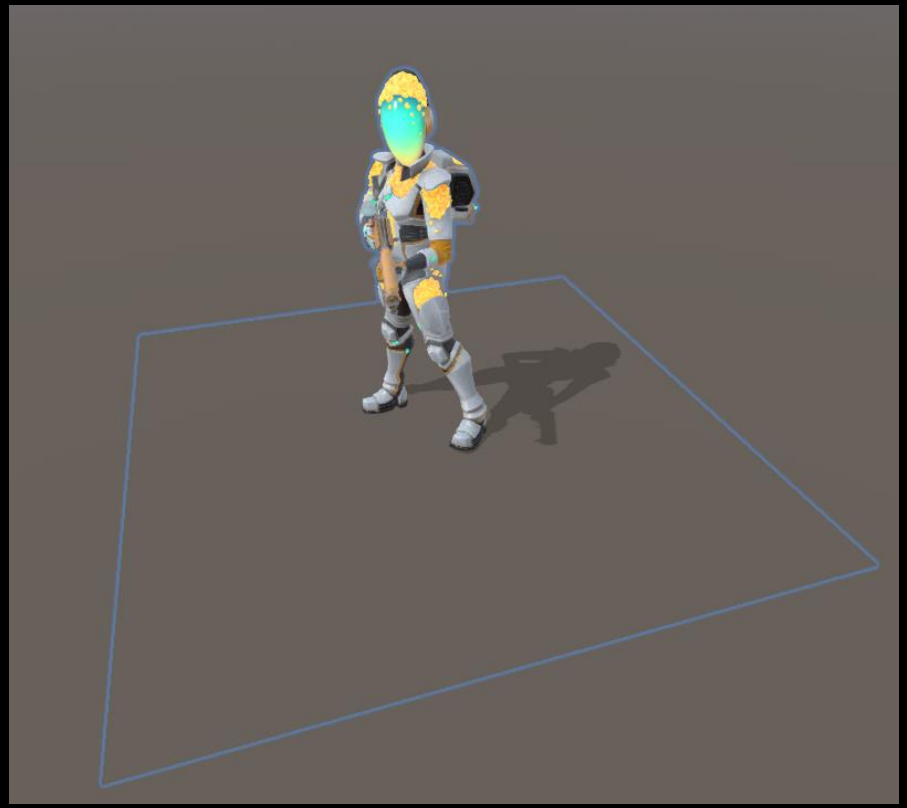
- Custom Shader for a Quad
  - Placed at base of AR Prefab.

- Shader Structure:
  - Render as Transparent
  - Include new LWRP Shader Libraries:
    - Lighting.hlsl & Shadows.hlsl

  Fragment Shader:

```
half shadowColor = MainLightRealtimeShadow(input.shadowCoord);

return shadowColor;
```



Demo - Plant Placement

- Detecting feature points or a plane
- Overlap sphere at possible placement point
- All plants have a collider

- Plants have blob shadows
- Use similar shaders to Terraformer

Resources

- **AR Foundation Samples**
  - github.com/Unity-Technologies/arfoundation-samples
- **AR Foundation Manual:**
  - bit.ly/ARFdocs
- **Unity for Mobile AR Solutions:**
  - unity.com/solutions/mobile-ar

Thank you!

Questions?

Ashley Alicea
Evangelist, Media & Entertainment - Unity Technologies
ashley@unity3d.com
@avashly