

# Fog Of War quick start Guide

There are several sections to this Guide:

- Importing the package
- Setting up your scene
- Adding/Customizing revealers
- Adding Hiders
- Customizing Fog Of War
- Debugging

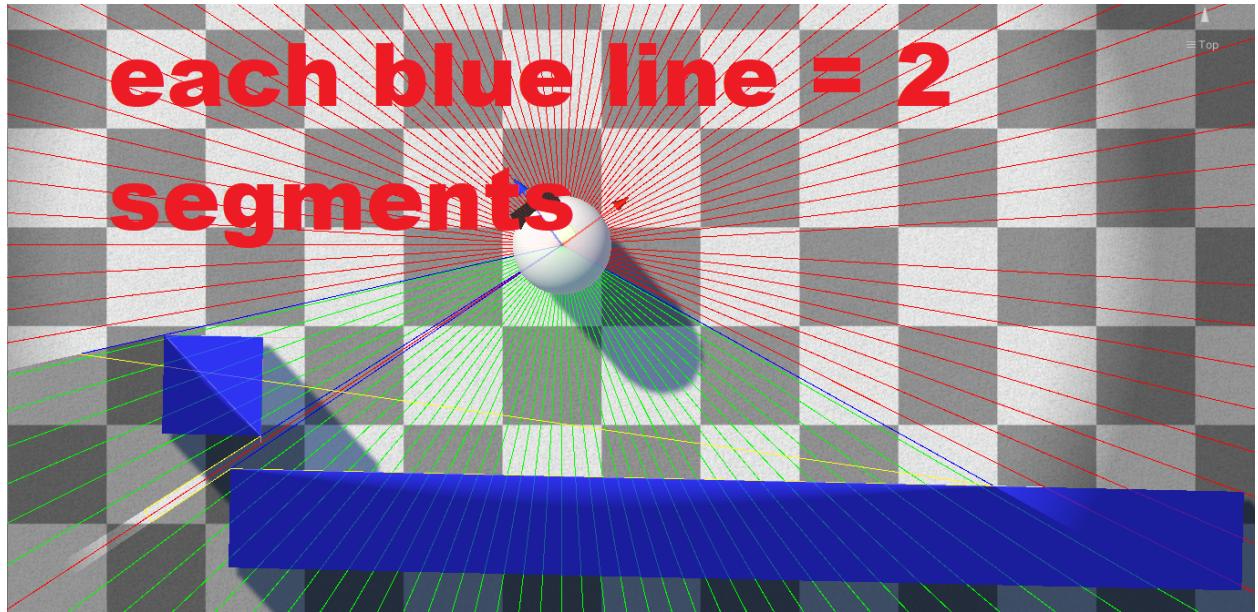
## Importing the Package:

- 1) Import the package via the package manager. If you are reading this, you have likely already done this.
- 2) If you are on URP or HDRP, install the respective packages, FogOfWar/FOW-URP or FogOfWar/FOW-HDRP. You can then delete the “Built-In (legacy) RP” folder.

## Setting up your scene:

First, add a new Empty GameObject to your scene. Name it “Fog Of War”, or something along those lines. Add the “Fog Of War World” component to this object. **There are a few variables here that need to be set properly to avoid errors**, that will change depending on your project.

- **Max Possible Revealers:** the maximum number of possible revealers active at once is capped because resizing buffers on the gpu is slow, so the buffer length is initialized with this value.
- **Max Possible Segments Per Revealer:** this value is constant for the same reason the previous one is. A “Segment” is defined by how many edges the revealer sees during its line of sight calculation. (in the figure below, 2 segments are added for each blue line). Setting this value too low will cause errors, so ensure you leave it at a reasonable value. To display the same debug lines you see in the figure below, enable Debug Mode on your revealer component.



In the Customizing Fog Of War section, we will go over the other variables you can change on this component.

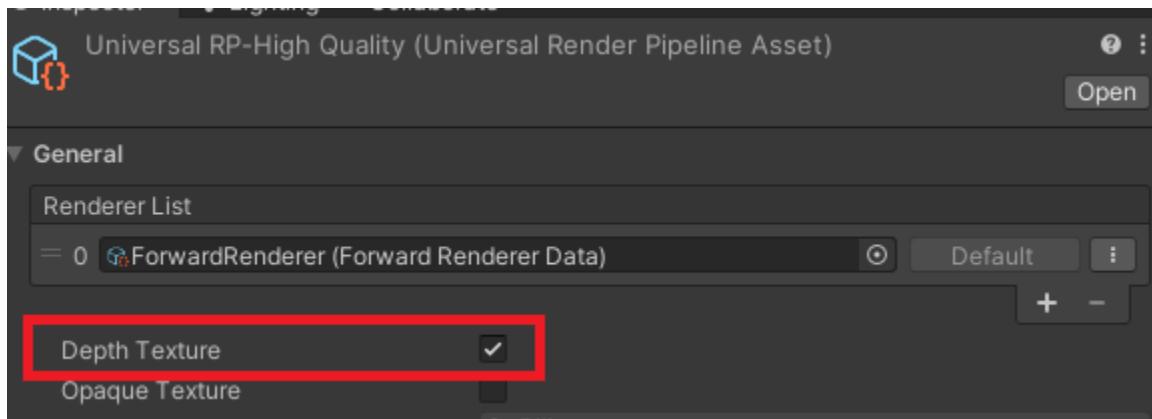
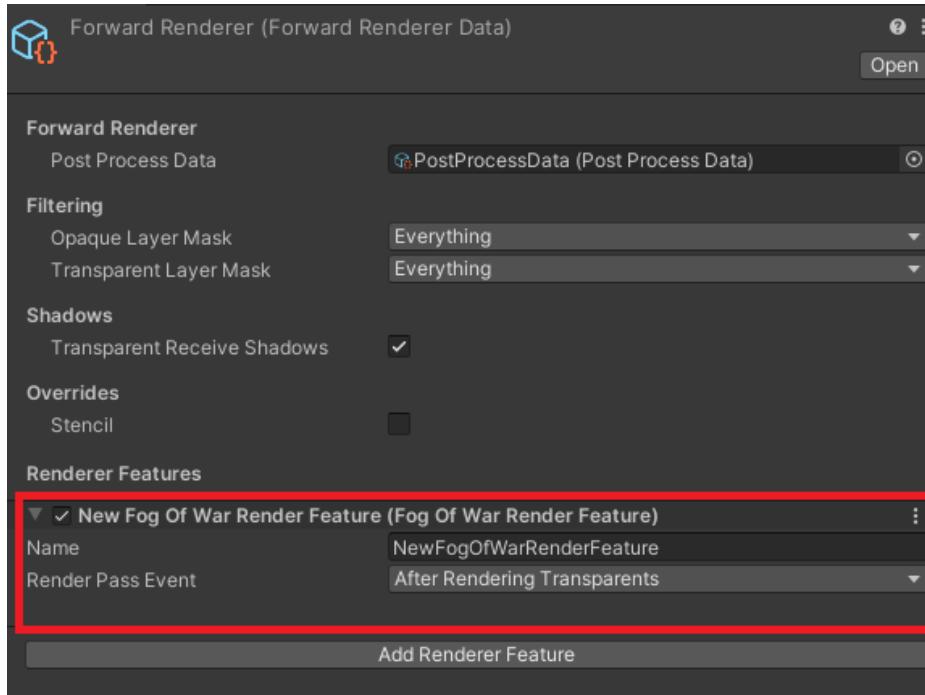
The rest of this section will differ depending on your render pipeline.

### **Built-In Render Pipeline:**

- Add the FOW Image Effect component to your main camera.

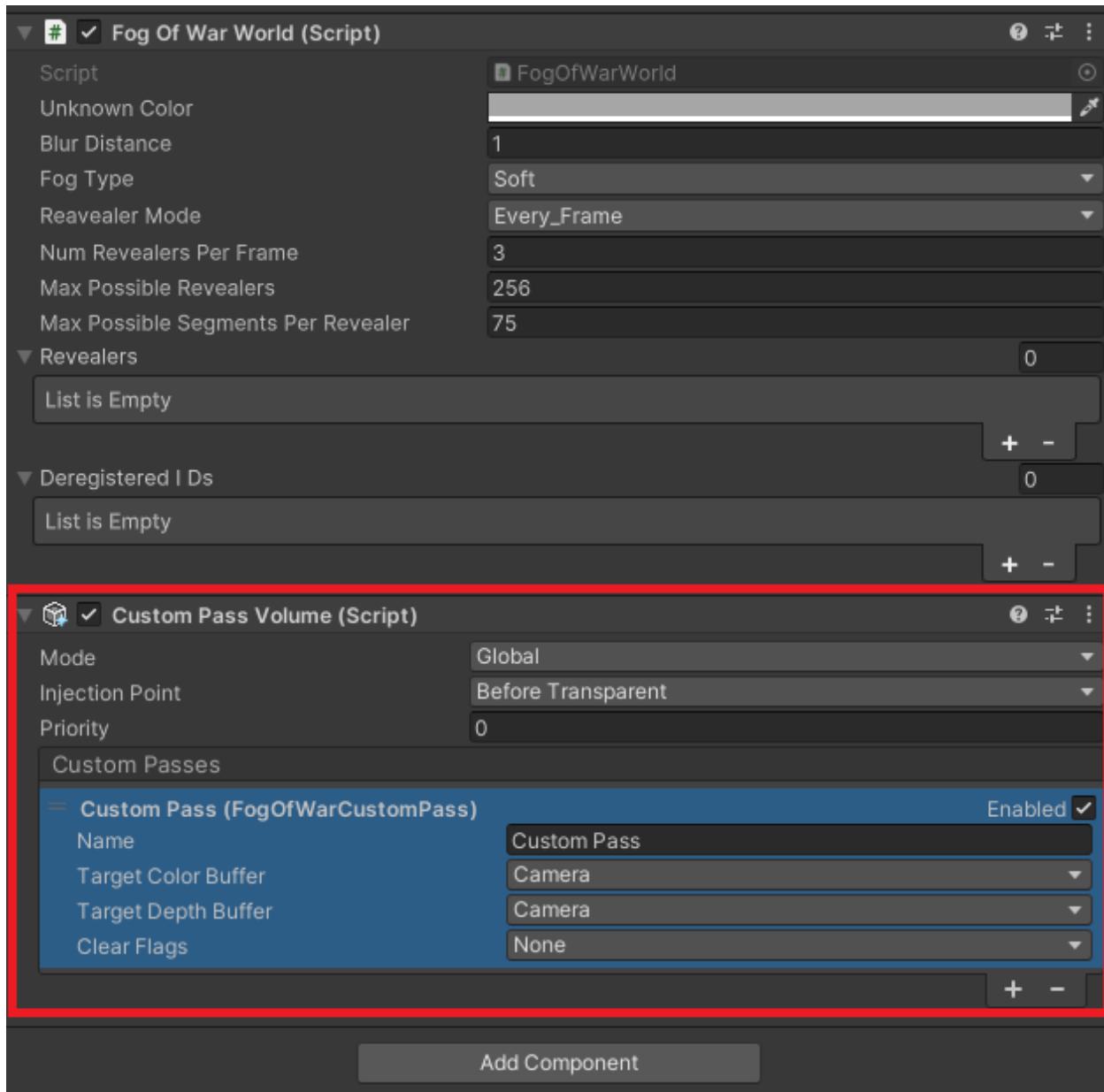
### **Universal Render Pipeline:**

- Add the Fog Of War render feature to your Renderer Data Asset
- Ensure “Depth Texture” is enabled on your URP assets



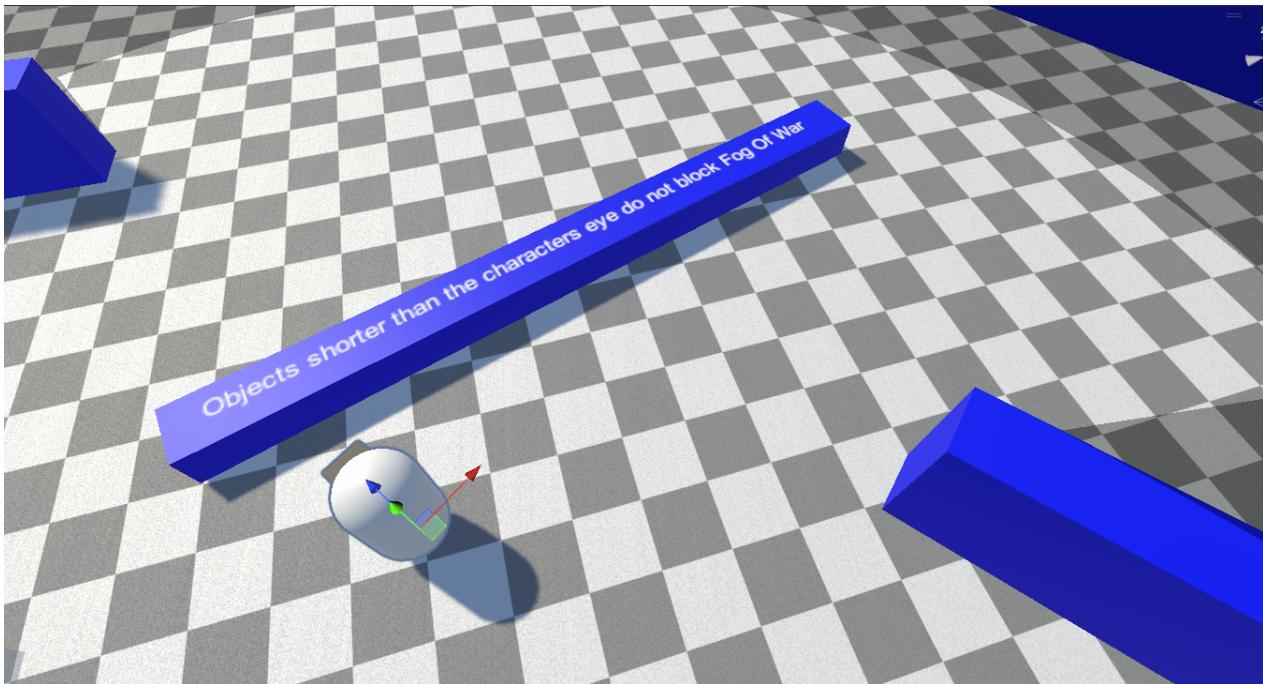
## High Definition Render Pipeline:

- HDRP is supported via the Custom Pass Framework. On your Fog Of War gameobject, add a new Custom Pass Volume component.
- Set the custom pass injection point to Before Transparent.
- Add a new Custom Pass to the list. Select Fog Of War Custom Pass



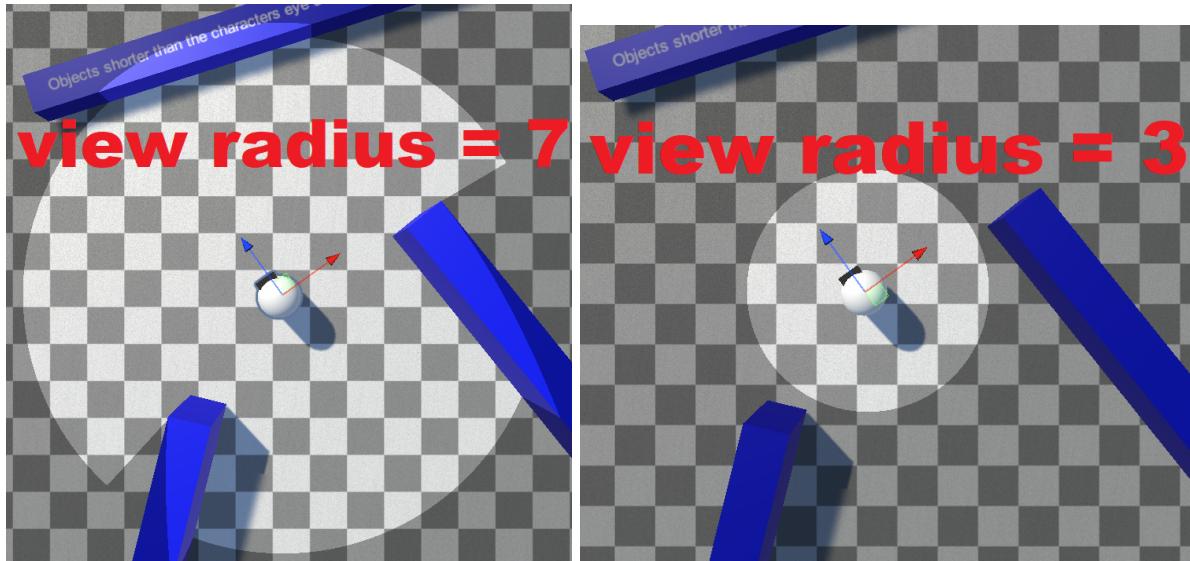
## Adding/Customizing Revealers:

Adding revealers is very simple. Simply add the Fog Of War revealer component to a GameObject. Revealers use raycasts to determine what should be in or out of the line of sight. Change the Obstacle Mask to the layers of the objects which should block a revealer's vision. **The raycasts will be performed perpendicular to the Y-Axis, on the X-Z plane.** This means that if a collider is higher or lower than the revealer's y position, it won't block the revealer's vision. (see figure below)



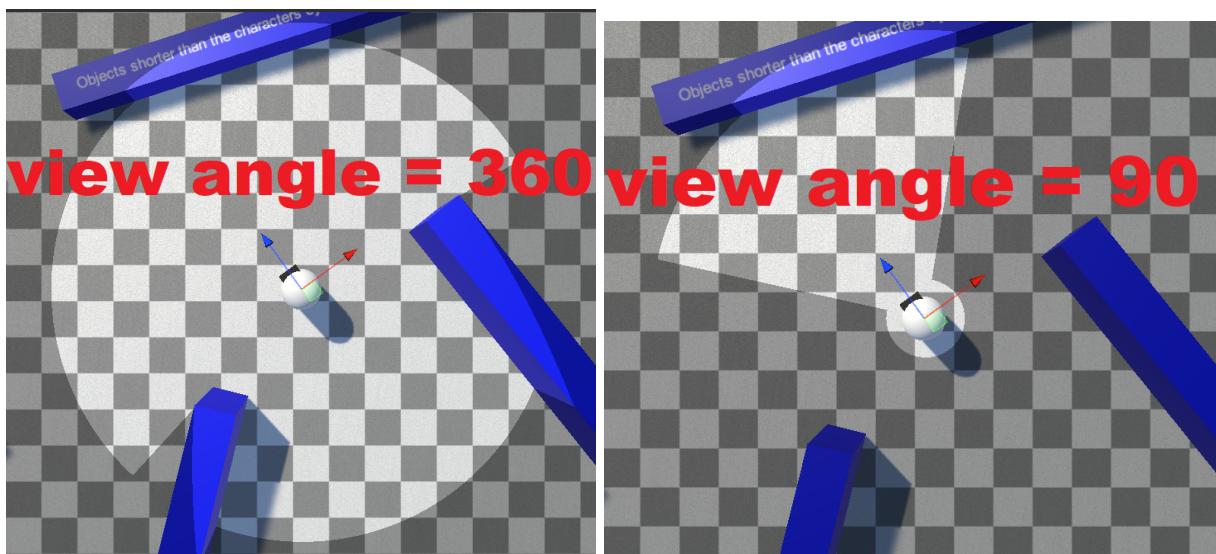
## View Radius:

View radius is how far away your revealer can see.



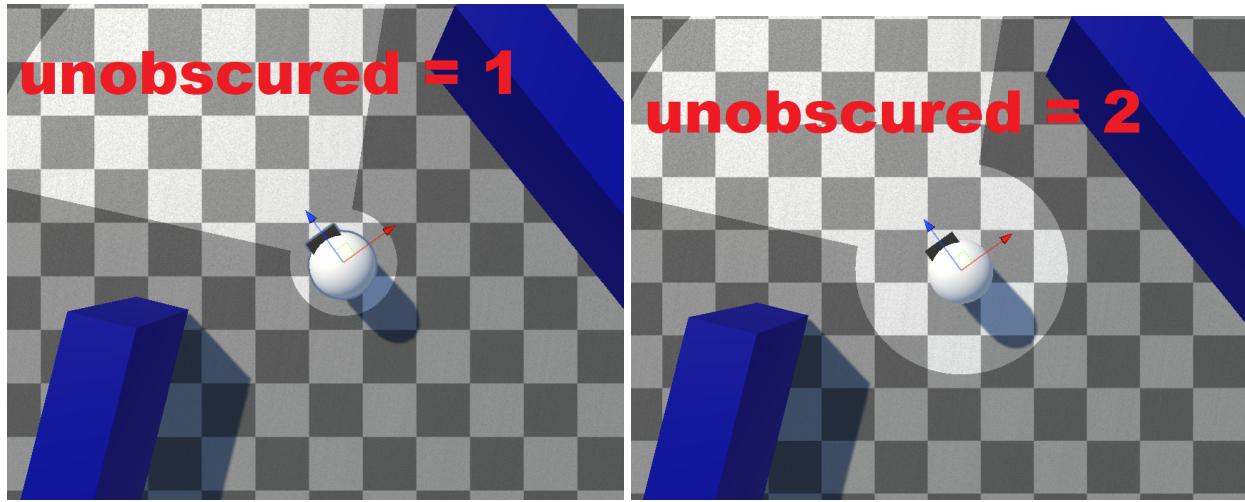
### View Angle:

The angle at which your revealer cuts vision off



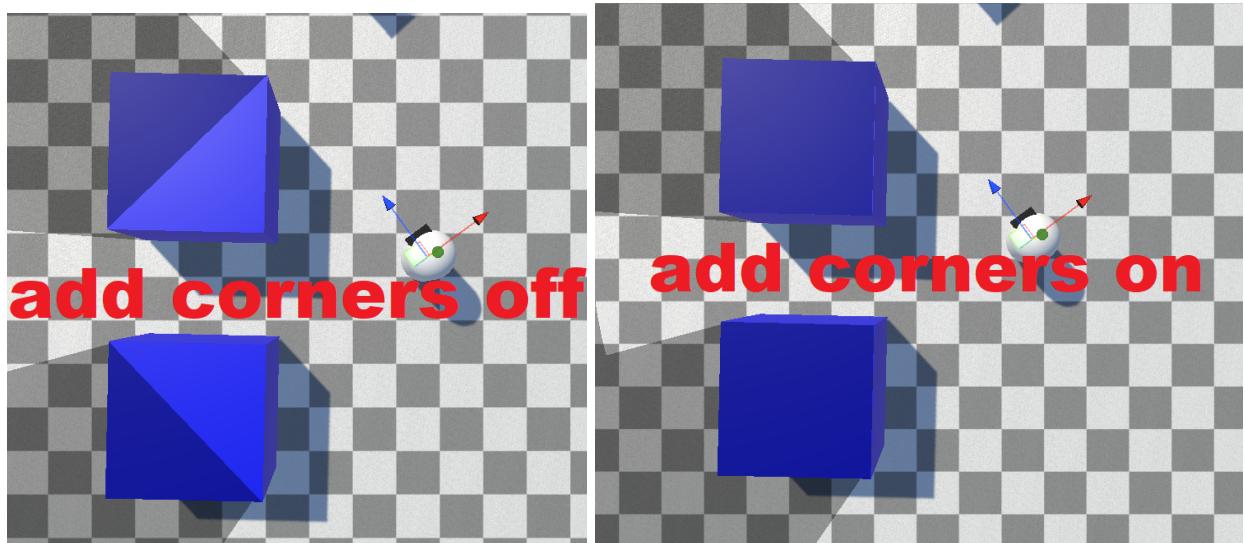
### Unobscured Radius:

The distance that should always be lit, even if its outside the view angle or blocked by the world.



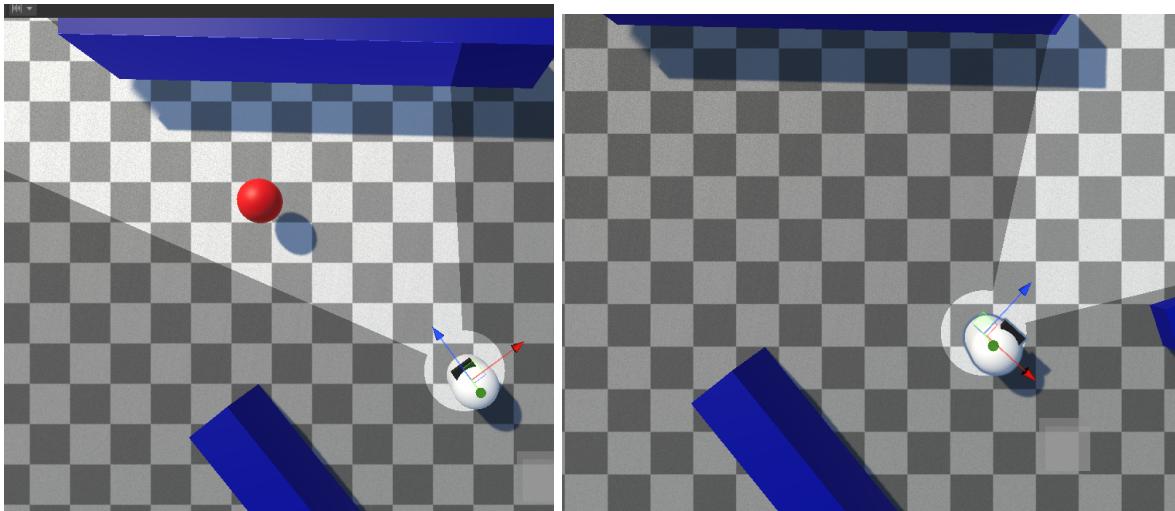
### Add Corners:

Determines if corners of objects should be added as segments.

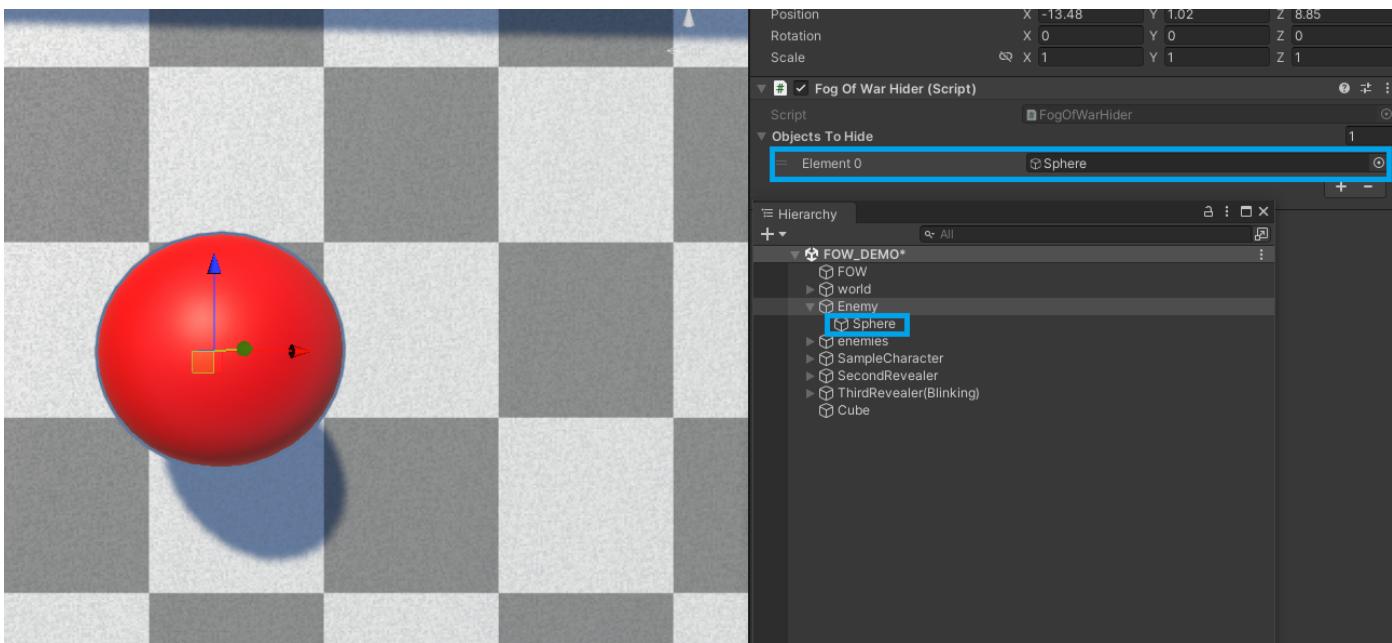


## Adding Hiders:

Hiders are objects that are hidden when outside of Fog Of War.



To add a Hider, simply add the Fog Of War Hider component to a gameobject, and add all the objects that should be disabled when the revealer cannot see that gameobjects position. (see figure below for setup)



# Customizing Fog Of War:

There are many ways you can customize your Fog Of War.

## Changing Color:

- Change the “Unknown Color” variable on the Fog Of War World component.

## Changing Fog Type:

This package comes with several types of Fog Of War.

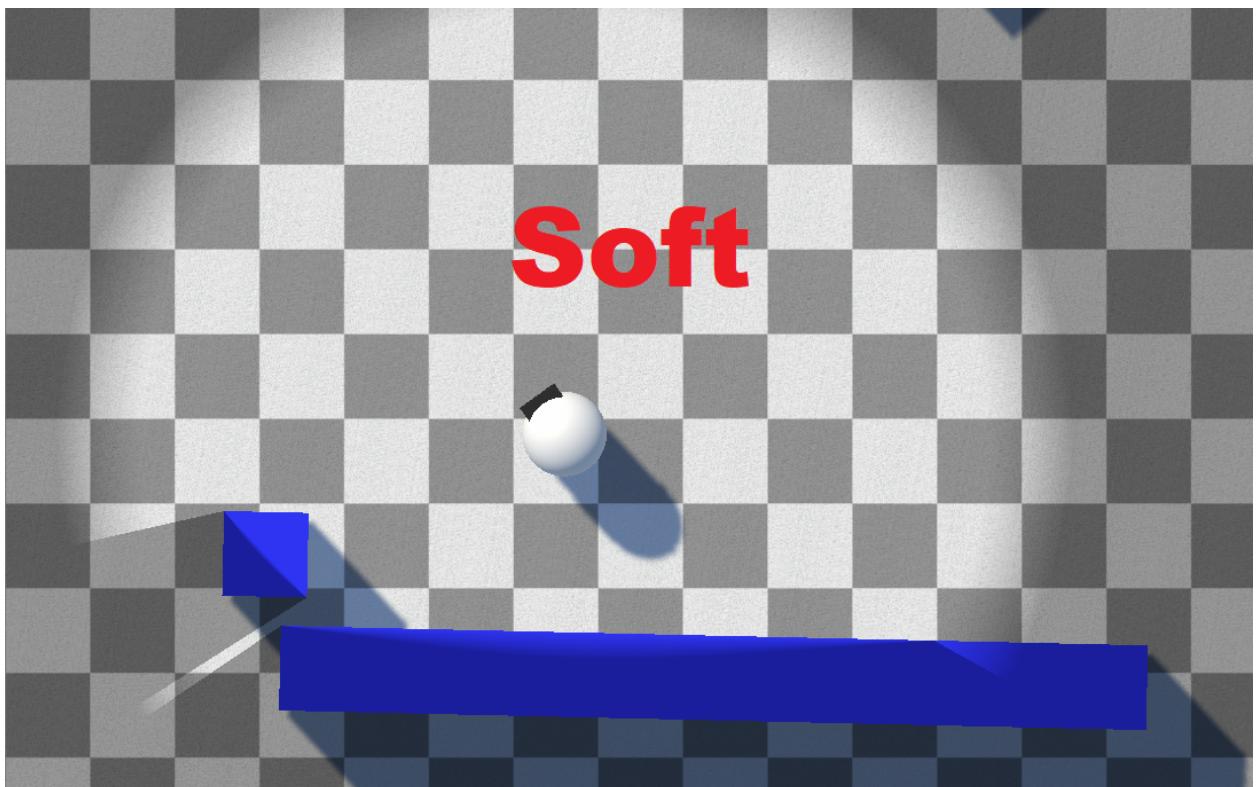
- No\_Bleed: cuts fog of war off without adding a cone to the end of the segments



- Hard: same as no bleed, but adds cones to the ends of segments



- Soft: blurs edges of segments for a nice gradient. Blur distance can be controlled with the “Blur Distance” variable.



## **Revealer Mode:**

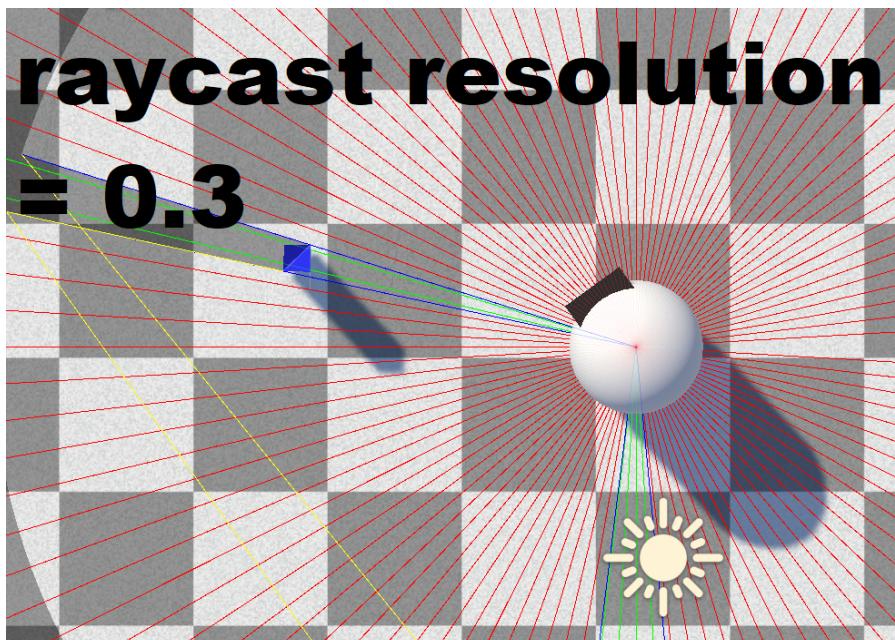
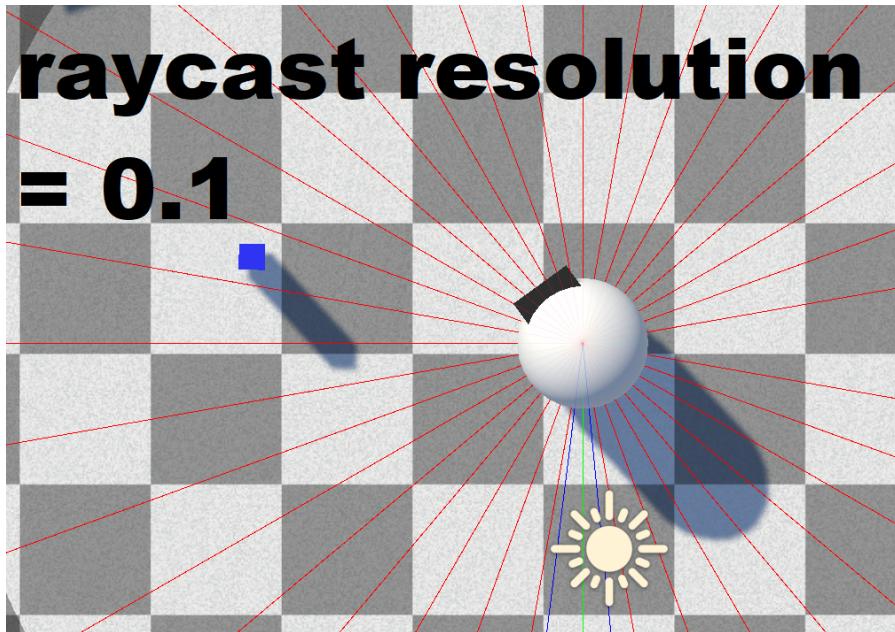
There are revealer modes with this system:

- Every\_Frame: Every single revealer calculates line of sight every frame.
- N\_Per\_Frame: Revealers will take turns calculating line of sight. Only the number of revealers defined by NumRevealersPerFrame will be rendered.
- Controlled\_ElseWhere: this setting will make none of the revealers calculate line of sight, perfect for if you want to set up your own logic for doing so.

## Debug:

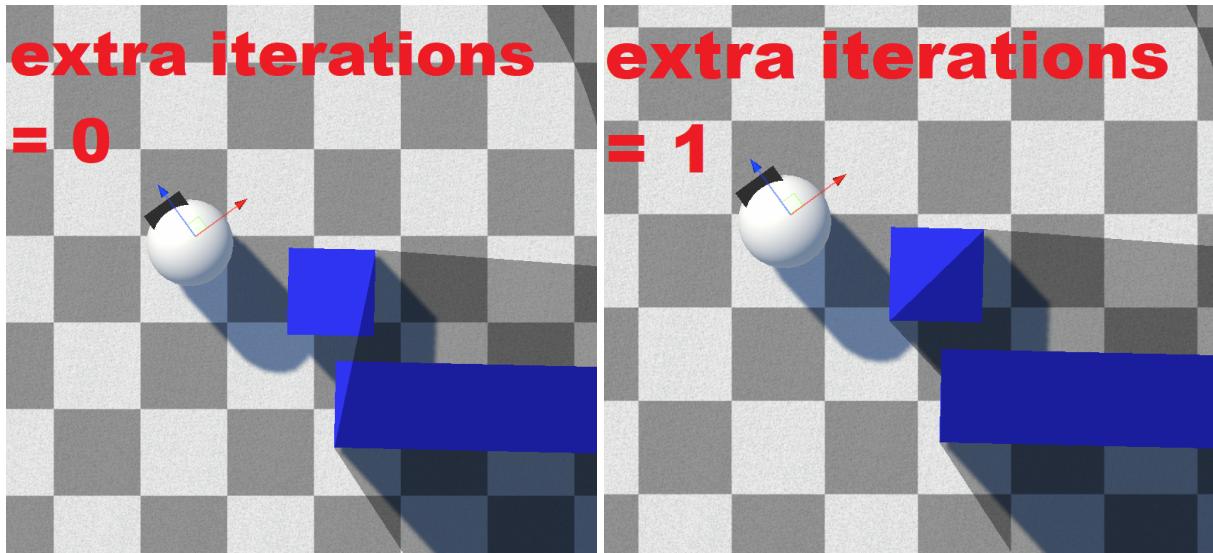
### My Revealer is missing objects:

Try turning up your RayCast Resolution on your revealer. The default is 0.3, which means that a ray is cast for about 1 every 3 degrees. If your object is small enough that it happens to be missed by these base rays, the revealer wont know its there, and wont try to find the object's edges. You can enable Debug Mode on the revealer to see these initial rays.



**My Revealer misses some edges:**

Turn up the number of Extra Iterations.



### **Additional Help:**

- Join the discord: <https://discord.gg/KSedK9UDn4>
- Fill out a contact form here:  
<https://milk-drinker01.github.io/contact/index.html>