Akıllı Mum

Visit for Perfect Assets: https://www.akillimum.net Please contact me on: akillimum@gmail.com

Follow on: https://twitter.com/ArVrUnity

https://www.youtube.com/@akillimum

Skype: hayletbisi

Discord: Akilli Mum #6862

Please review the packages even if you do not like them. And please send me an email first for the errors :) I will solve it for sure ;)

Standard Reflective Shaders 4.x

updated on - August 6, 2021

IMPORTANT: I am re-writing the reflective shader package for Unity 2020.2 and later with cleaner and faster code (also solves the lag problem on VR headsets - thanks to **SJAM** for his help).

So I am writing a **“MirrorManager”** script to manage the new reflections. If you are using a smaller version than Unity 2020.2 or non-VR; you can still use previous docs and scripts. Please refer to old doc here:

https://docs.google.com/document/d/1We78qz0Fa54B3fepdrEbr2oAG4SHIA0TTA3qsLBSiHM/edit#

I will not remove the old “CameraShade” scripts from the package. So as i said, you can use them too for non-VR apps (until i finish the whole implementation with new “MirrorManager”).

So what this version includes:

It only has a full mirror and reflective surface (also transparent glass like) implementation right now! Other things (mirror in mirror) not implemented yet!

If you do not target VR or using Unity below 2020.2:

I will implement all script-shader properties soon for this version (WIP), but until that time you can use old scripts and shaders as I mentioned before. Here is the previous docs:

https://docs.google.com/document/d/1We78qz0Fa54B3fepdrEbr2oAG4SHIA0TTA3qsLBSiHM/edit#

Known Bugs and Issues:

* Reflection flickers if “Temporal AA” is selected

Reflection flickers if “Temporal AA” is selected with Post Processing. Other MSAA solutions seem to work normally.

Limitations:

As version 5.01 all VR pass modes works :) for 2020.2 and above.

Users are telling other VR devices are working fine too (like Steam, HTC); but i can not guarantee because i do not have any of those devices!

1. VR can not draw skyboxes. But I wrote custom shaders to mimic the skybox and to be reflected correctly on the mirror. (Standard Skybox, VR-Stereo Cubemap Skybox and VR-Stereo Equirectangular Panorama Skybox are implemented). If you are using some other skybox solution and can not get it to work please write to me :) Please look under the sample scenes VR folder for a sample fake-sky.

Please see below video to understand how to use (setup) a fake skybox on VR project (It just uses a big cube or sphere to create an illusion of a fake skybox object covers the entire scene):

https://youtu.be/Ft3q1qmrlOI?t=251

1. VR implementation will not support recursive (mirror in mirror).

If the package does not update correctly?

Package may not update correctly on 2020.2.x and later, so please follow below instructions for upgraded projects:

1. Backup :)

2. Delete AkilliMum folder and close the project

3. Goto your asset download folder and delete all (or u can just delete mine named as "Akll Mum"). Folders are here:

macOS: `~/Library/Unity/Asset Store`

- and
- `~/Library/Unity/Asset Store-5.x`

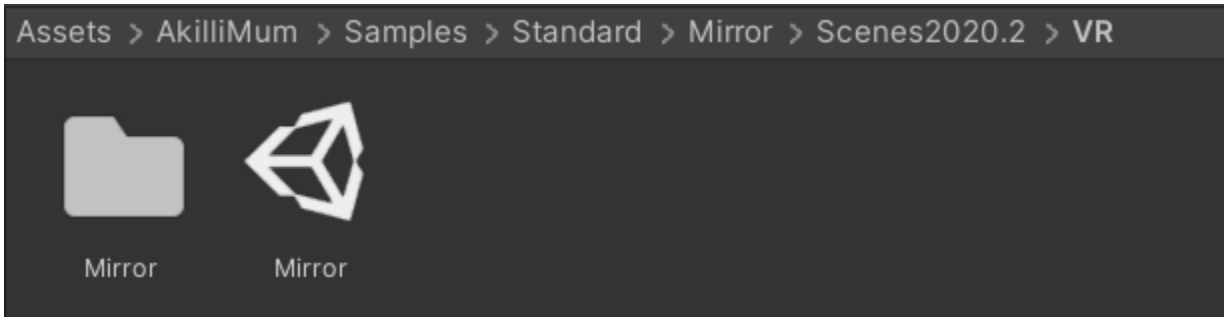
Windows: `C:\Users\{accountName}\AppData\Roaming\Unity\Asset Store`

- and
- `C:\Users\{accountName}\AppData\Roaming\Unity\Asset Store-5.x`

4. Open the project, download and import again from the package manager :)

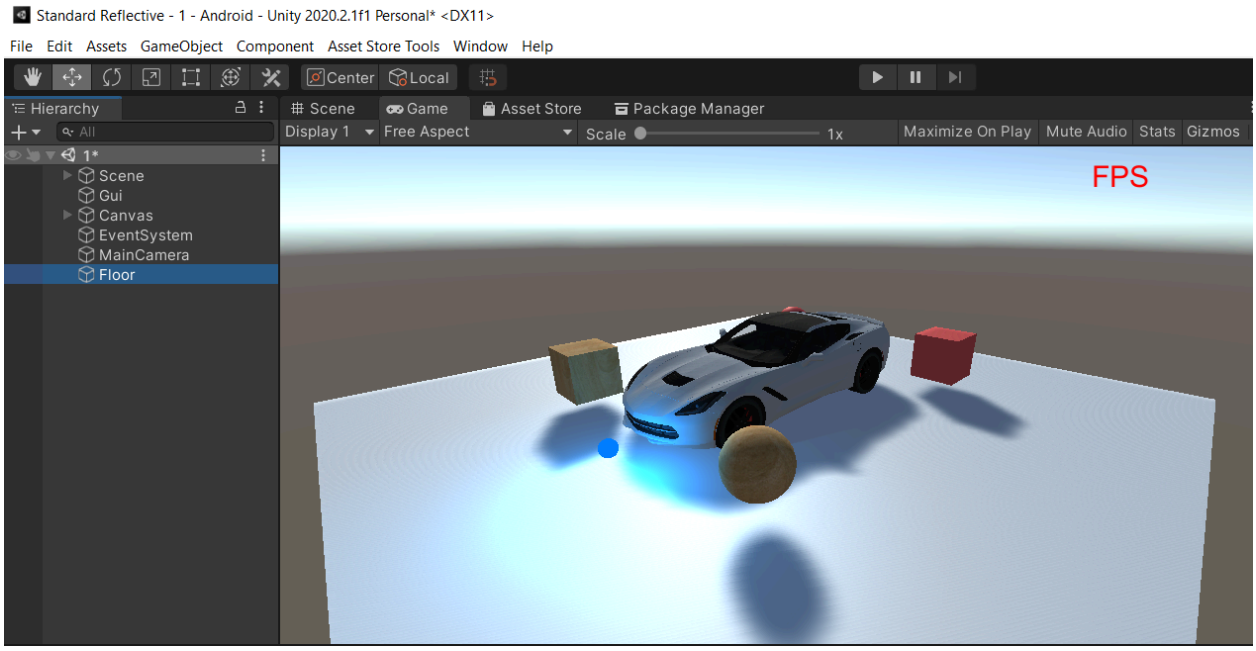
VR Samples:

I will put some VR samples under this folder (you have to install the XR plugin and make proper settings of course before running them :)):



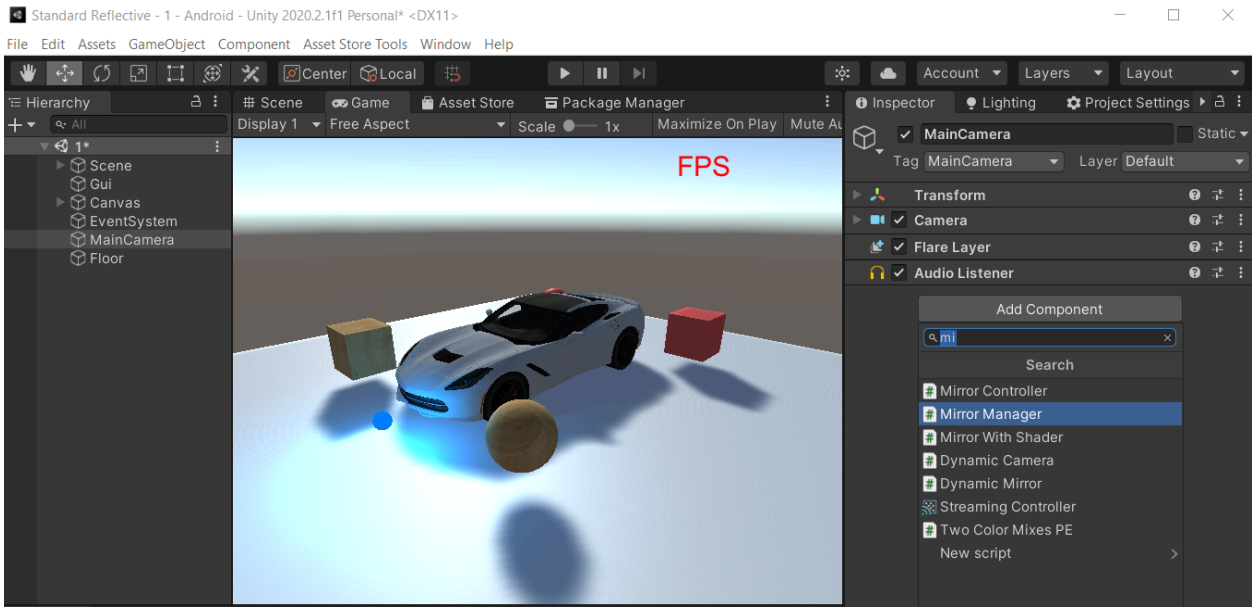
Setup:

Setup is pretty simple. Here is a simple scene with some objects and a floor:

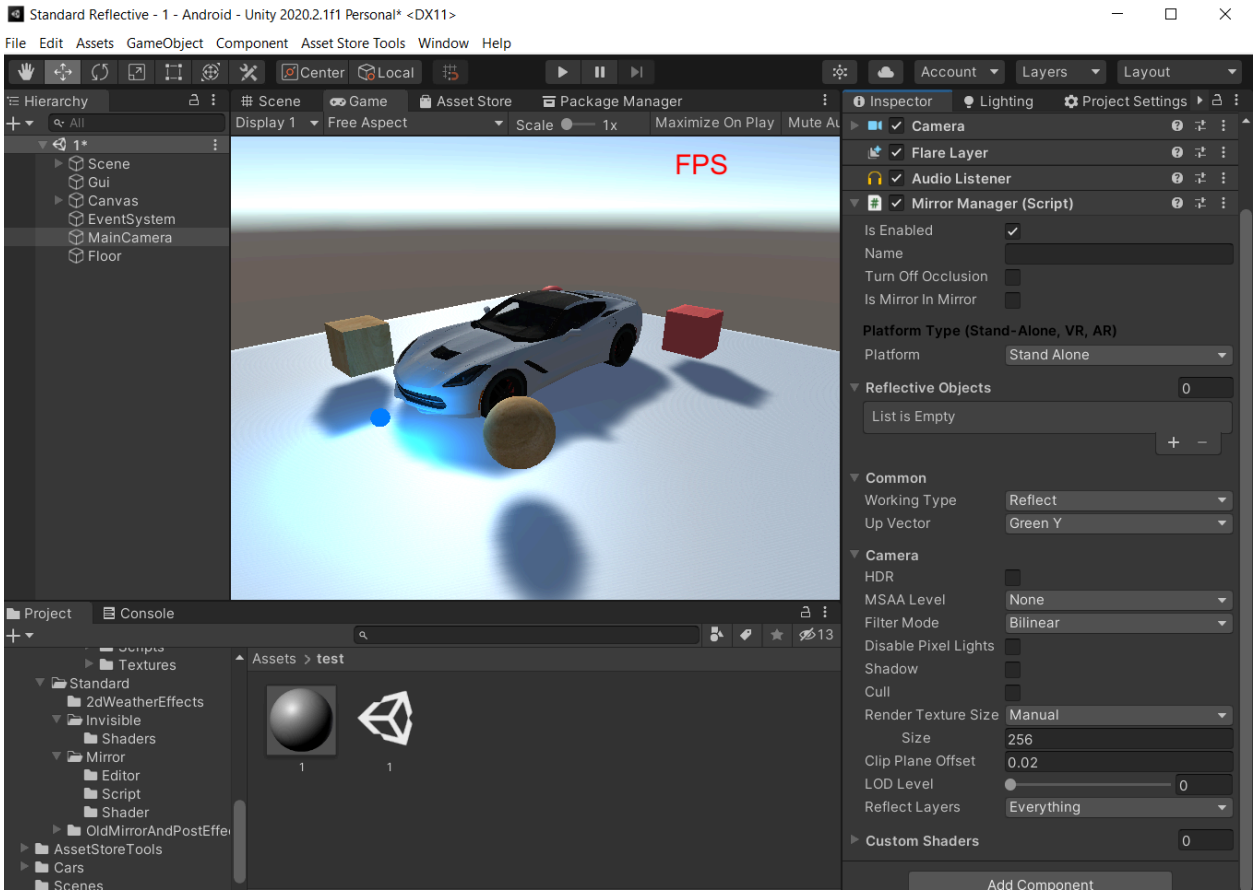


For this sample, I want to make the floor a full mirror!

Let's add our script to the main camera. Click the “Add Component” button when the camera is selected on “Inspector” window and search for “MirrorManager”:

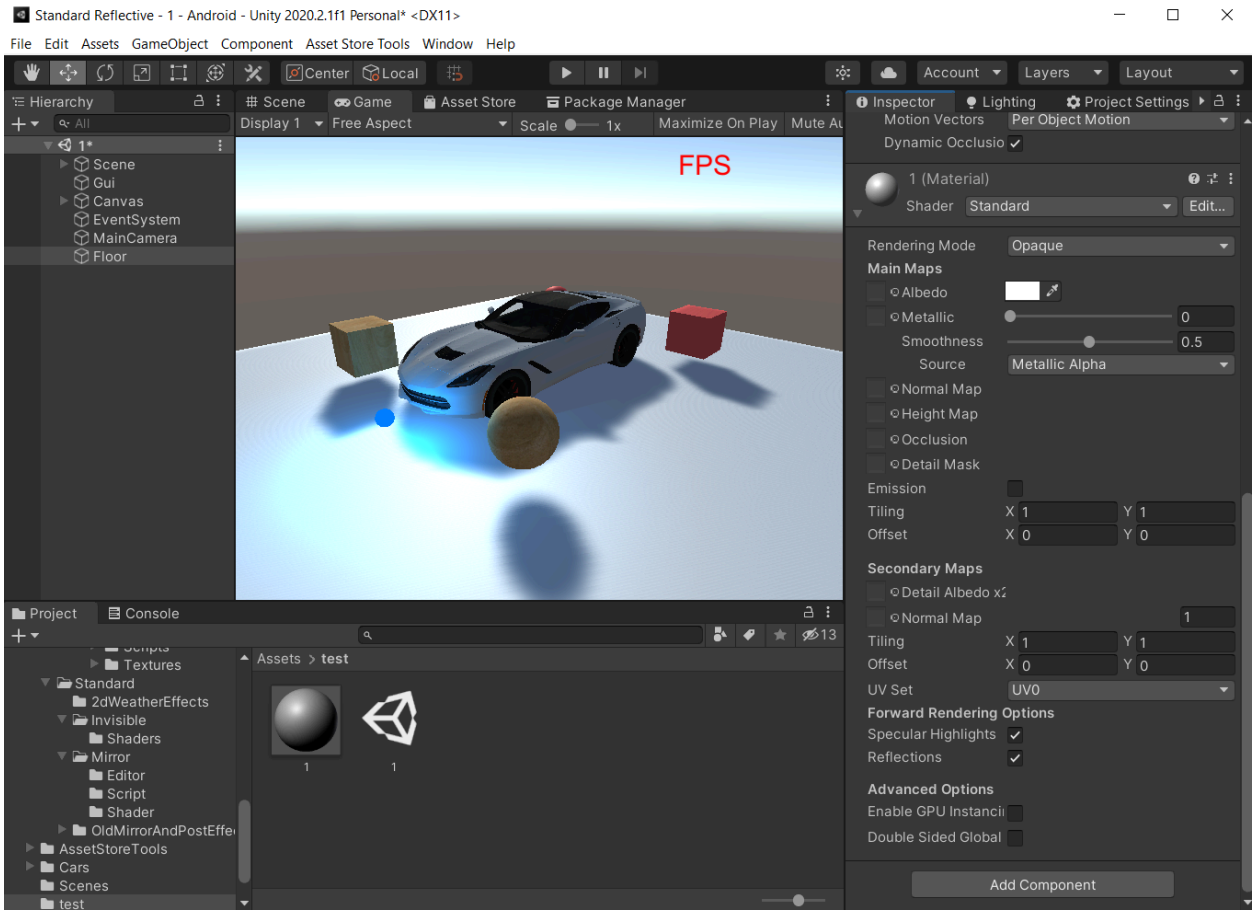


So our script will be added there:



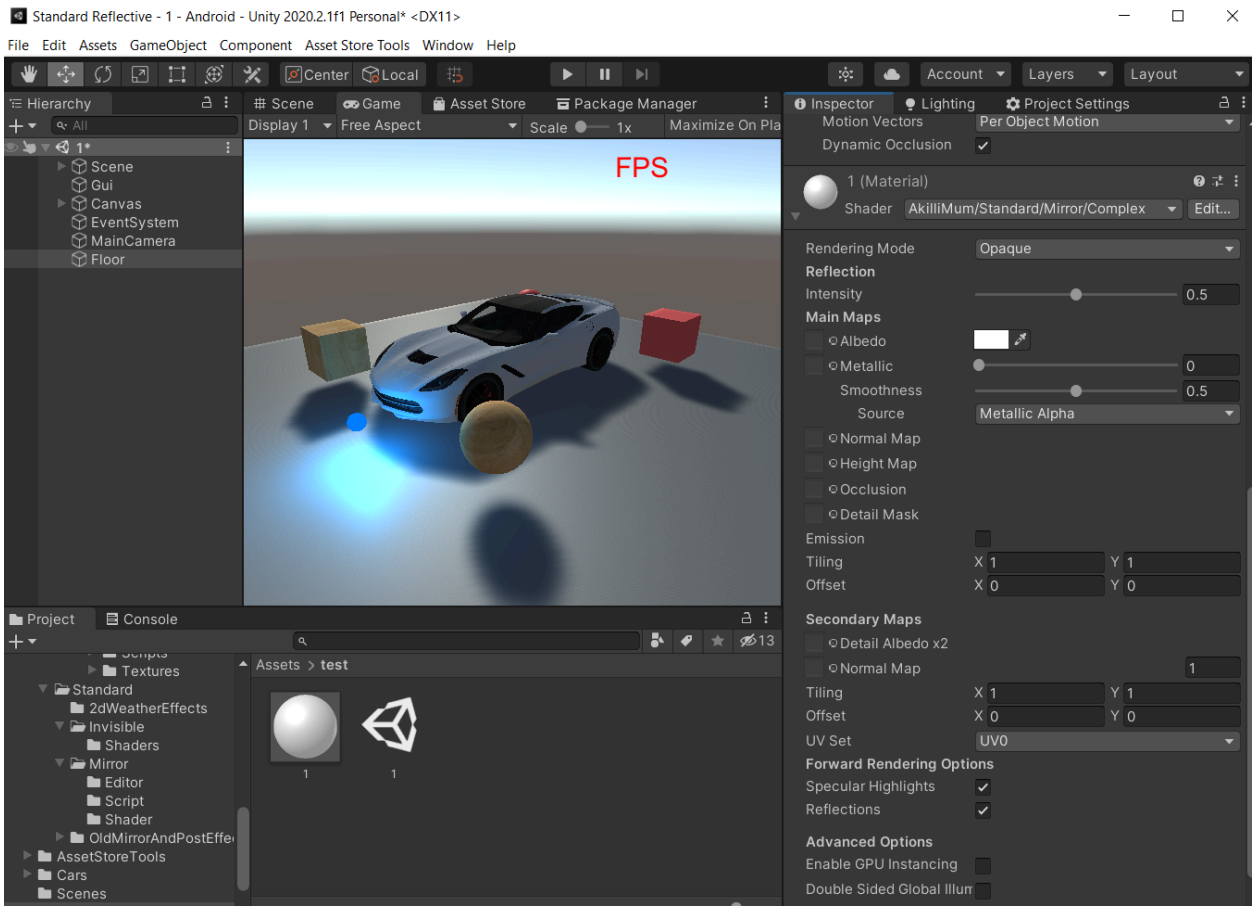
We will come to the properties later :)

Lets select our mirror (floor in this case) and set it up. First select the floor in “scene view” and change the its material’s shader:



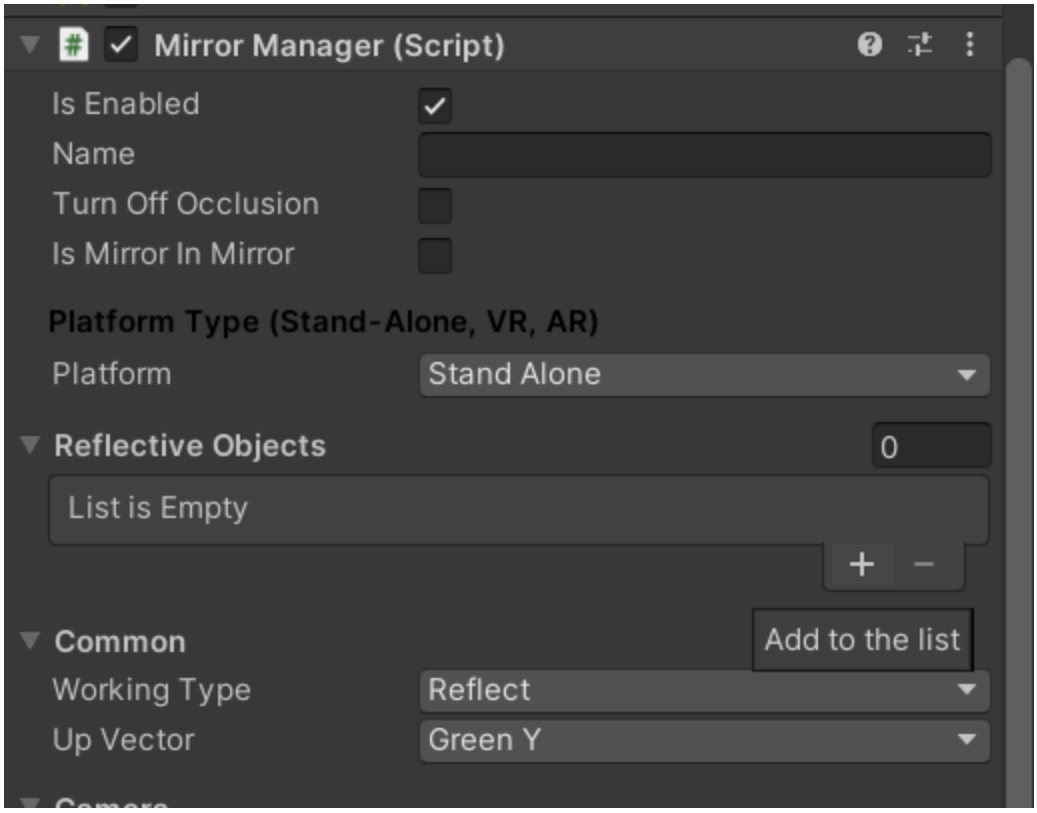
As you can see it is default standard shader, so we will select our one which is:

AkilliMum->Standard->Mirror->Complex:

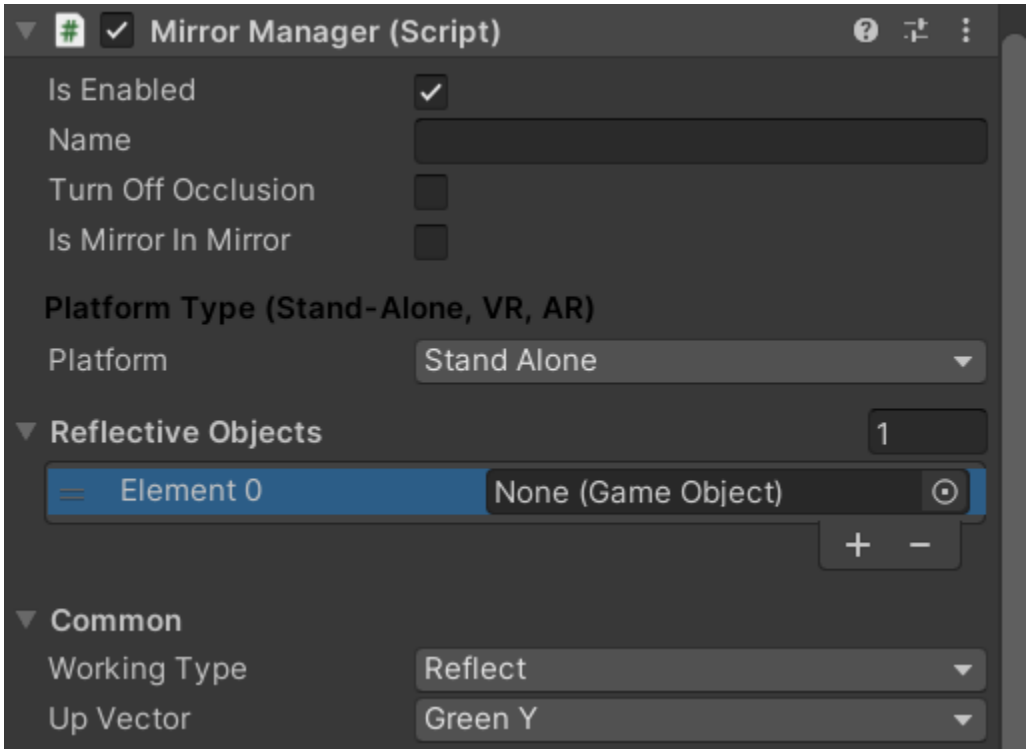


So as you can see it has all standard material properties and some others for our reflective extras.

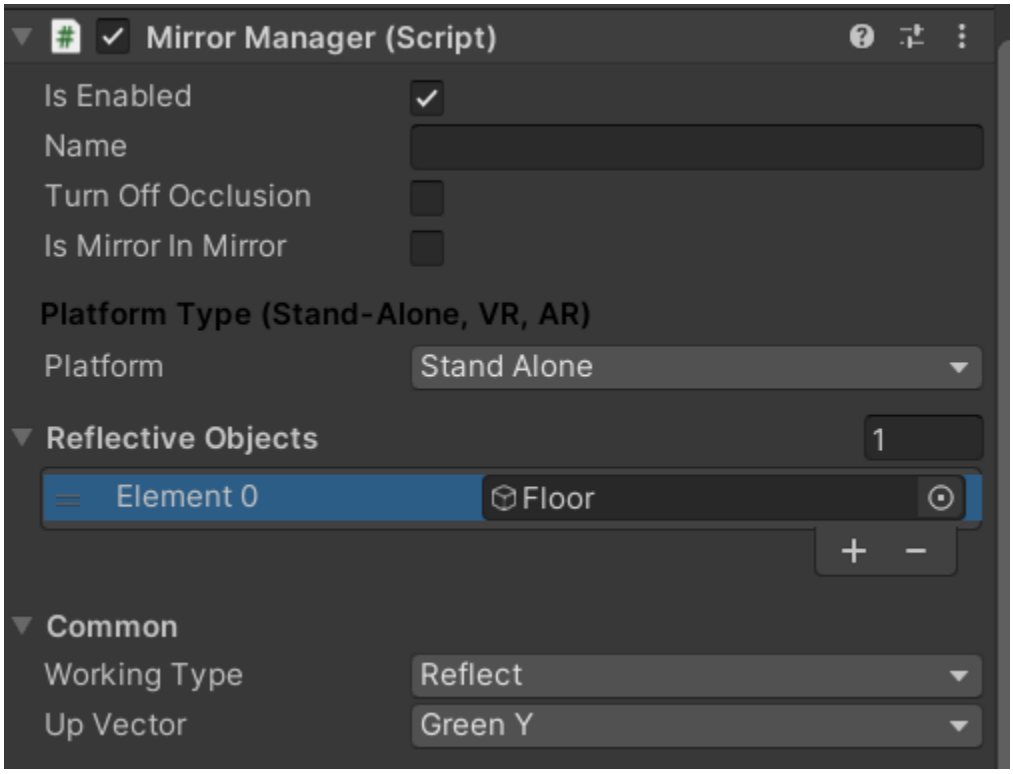
Let's go back to our script “MirrorManager”. We have to attach the mirror object (floor for this case) to the manager. Just go to the “Reflective Objects” list property on the script, open it and add an item:



Just click the + icon and it will add an empty slot:

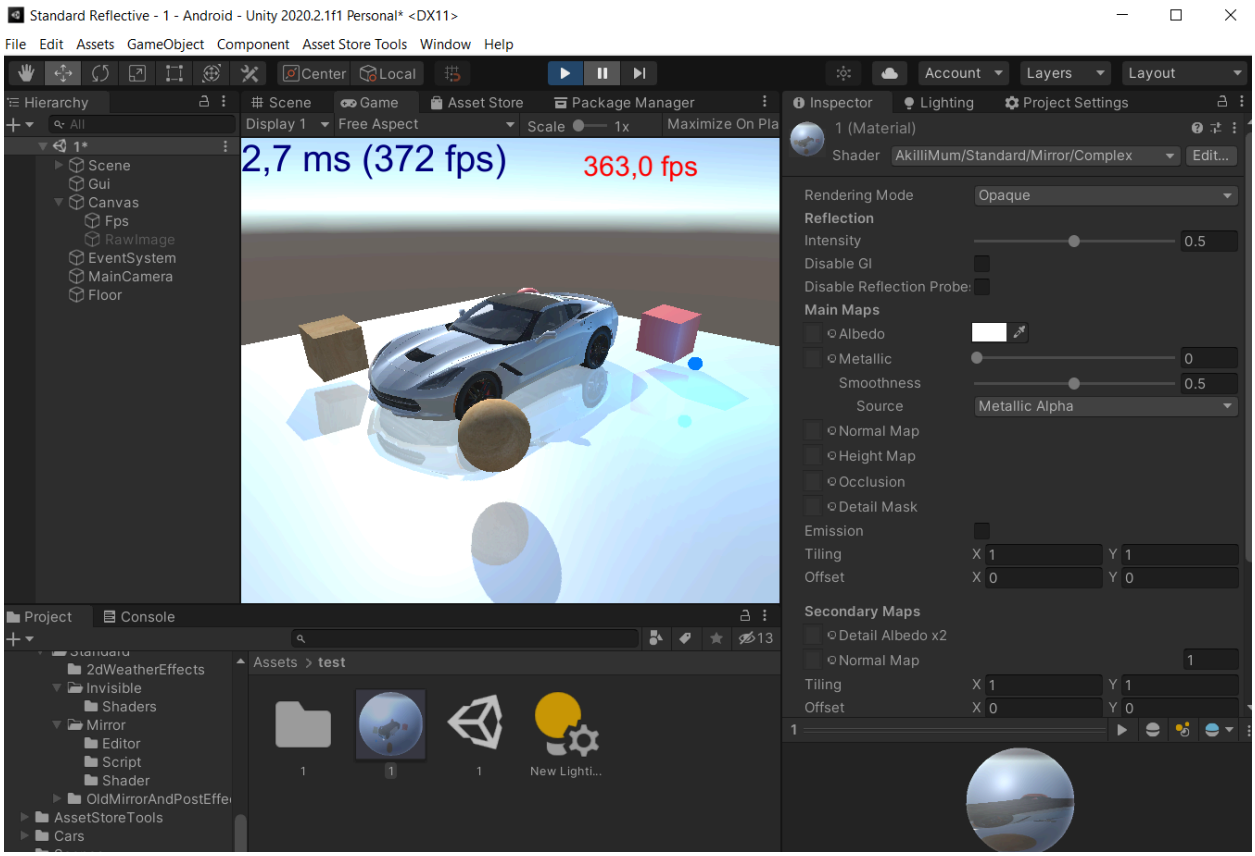


Now hold the mouse down (not release) on floor object on “Hierarchy” window and drag and drop here:

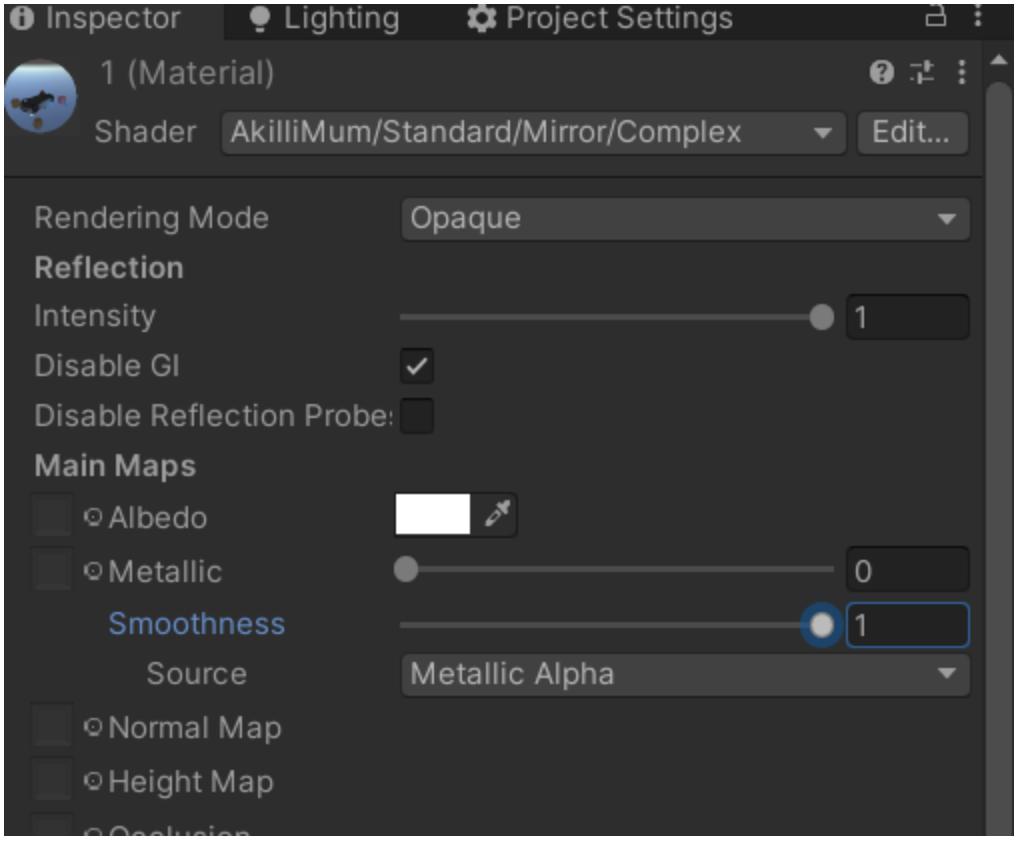


So our “Floor” gameobject will sit there.

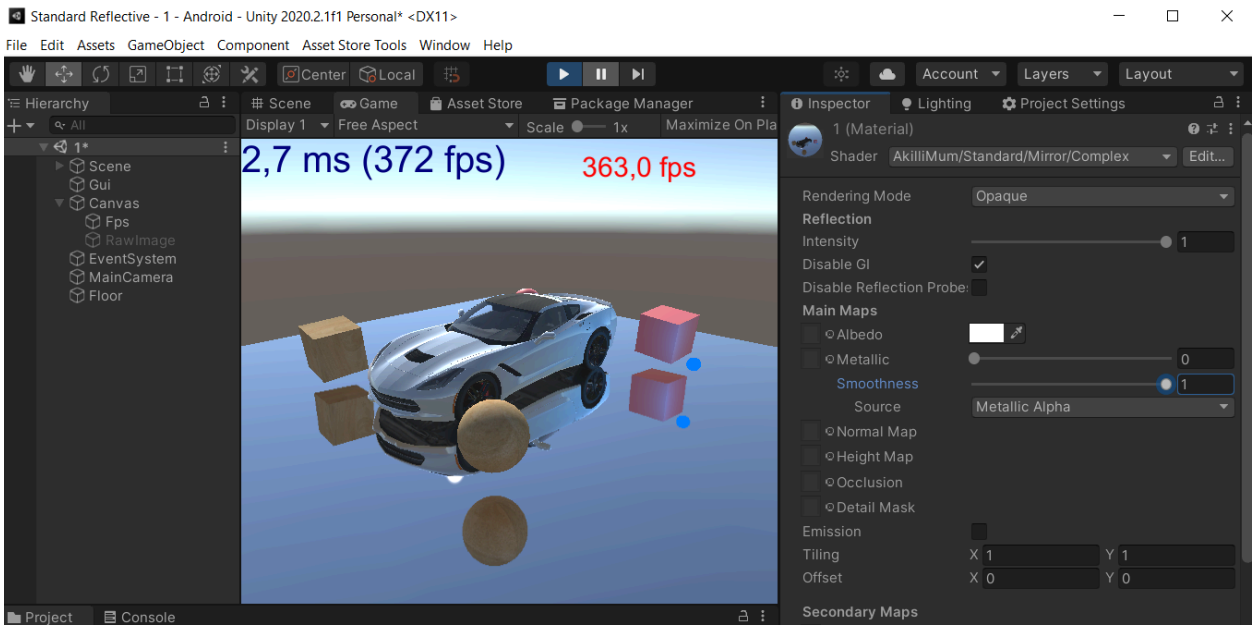
That is all, let's hit the play button :)



There is a reflection but not perfect? Because there are a lot of parameters on the shader to make it perfect or just a reflective surface. For example

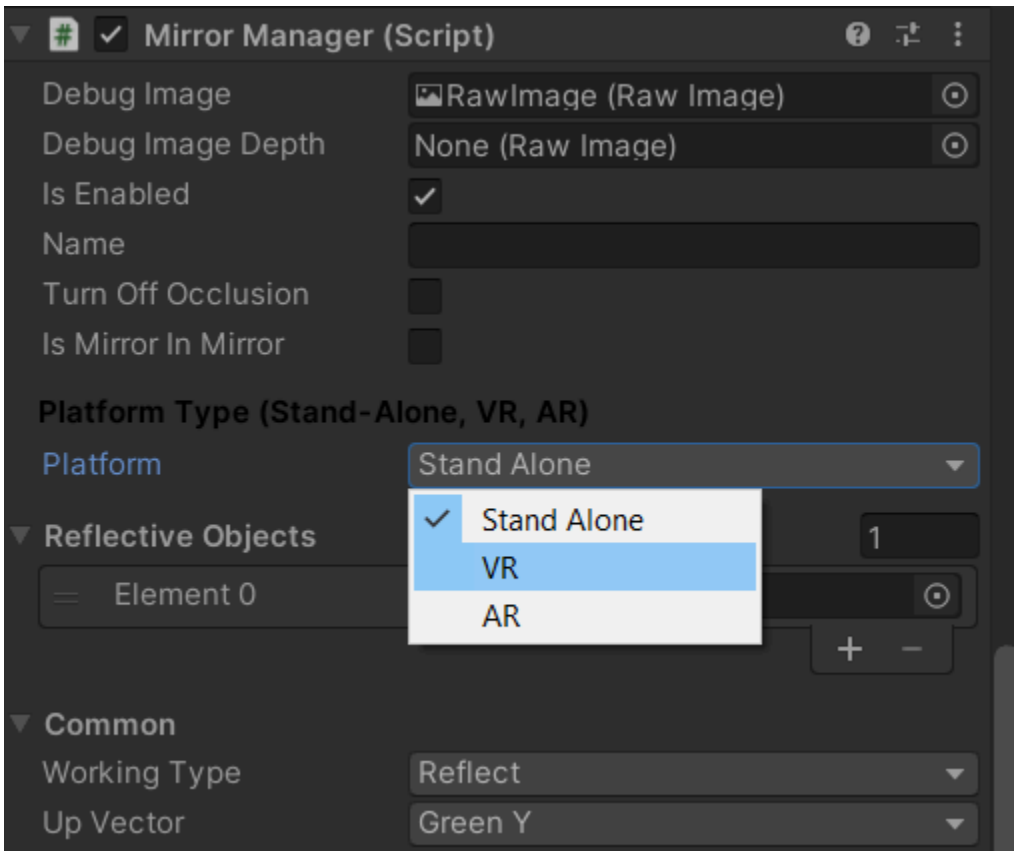


Make “Intensity (Reflection) = 1” “Disable GI” and set “Smoothness = 1” to create perfect mirror:



VR Setup:

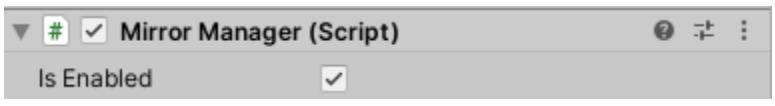
The VR setup is the same as the setup above. Just select the correct device from the “MirrorManager” (VR for this case):



I do not go deep into the VR setup because it is related to Unity and you have to know setup and run VR knowledge.

Project includes a “VR Samples” folder as I described above. So you can find the samples there.

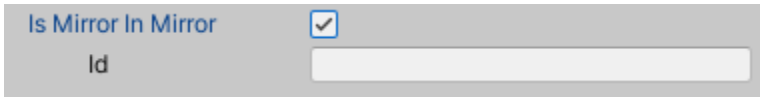
Mirror Manager Properties:



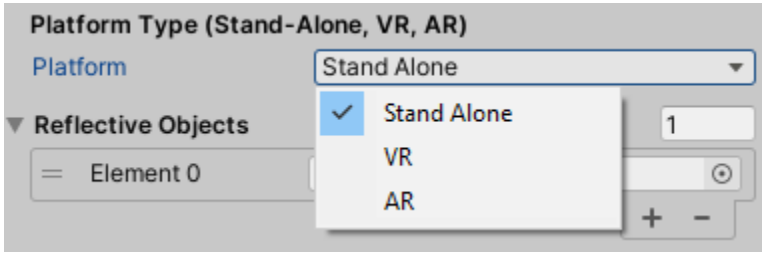
Use this to enable/disable the script (do not use main check on the script itself!). So you can stop-start the reflection as you want.

You can name the scripts to understand what they are made for. If you have several mirrors (yes you can create more than one :) Just add another “MirrorManager” to your camera and setup like above for your other mirror) you can forget or can not remember why it is used easily. So it is just a string for you to enter anything.

According to camera settings Unity may occlude some objects on mirror render (baked occlusion etc.). So it creates glitches on the reflection (some objects will not be drawn etc.). To disable that you can close the occlusion on the mirror camera.

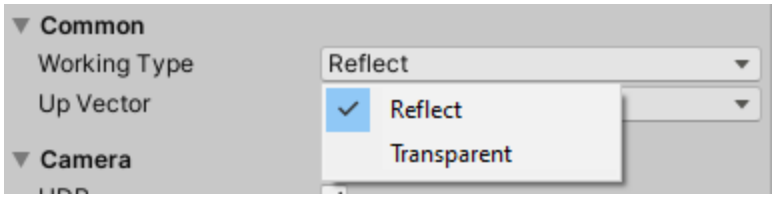


It will be used later to draw recursive (mirror in mirror) mirrors later. This version does not work but you can use previous scripts as I mentioned above. It should be disabled for normal mirrors!

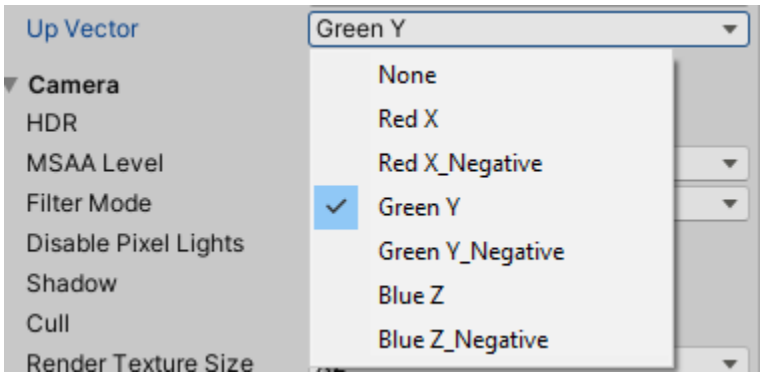


Platform Type: Platform is the main target of the device. So you have to select the correct dropdown item. As you can understand, “Stand Alone” stands for non-VR devices like PC,Mac,Mobile etc. while VR targets VR headsets and AR is special for AR apps.

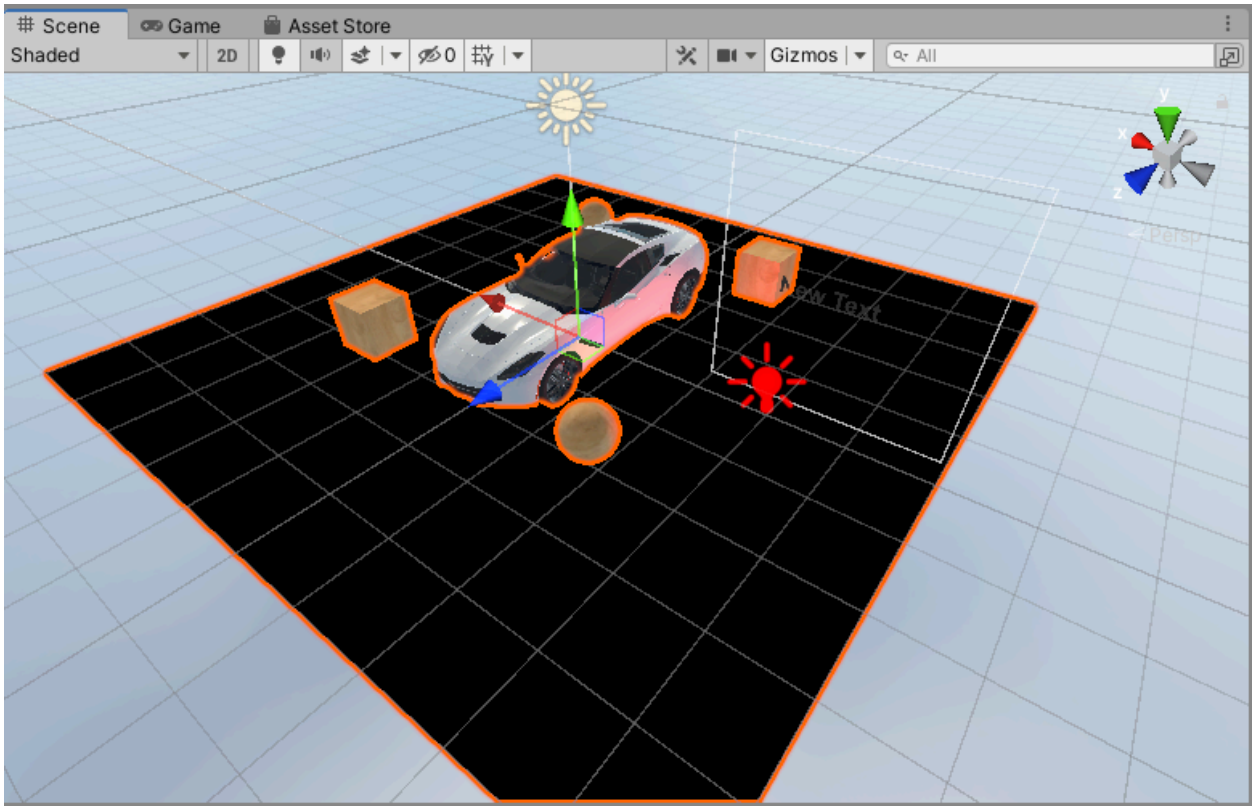
This is the list where mirror objects will be held. Mostly there will be one item (the mirror game object), but if you are using the script for the objects in the same plane direction (like wc mirrors which are all on the same wall), you can just add them here. So the mirror manager will draw the reflection only once and use the same mirror texture on all of them. So you will gain performance.



Working Type: This is how the mirror manager draws the reflection. Just now only “Reflect” is active and as you can understand it mirrors the scene. Transparency will be used later for AR and mixed (masked texture blending etc.) cases later.

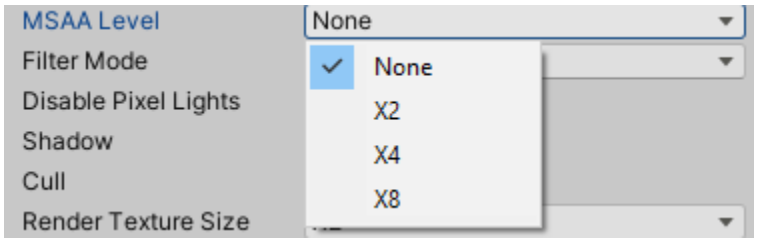


Up Vector is pretty important for the mirror to work. Most of the time the mirror object’s facing direction is “Green Y”. What that mean is, it is the direction of the object when you select the object on scene view like this:

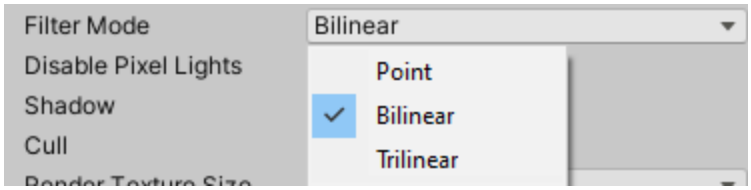


As you can see the floor object's Up Vector (facing the direction where we want the mirroring to happen) is the GreenY vector. But sometimes it can be different (because it can be a sub object in another object, or it may be imported from a 3d modelling program and its vector can be different). So you have to check it and select the correct Up Vector!

Should the camera draw with HDR?



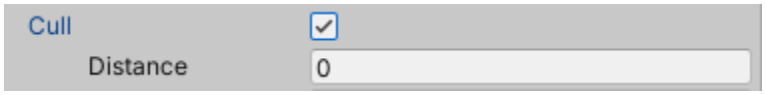
MSAA (Anti aliasing level): It can draw the reflection as anti aliased, but it may decrease the performance specially on mobile platforms. So use it carefully!



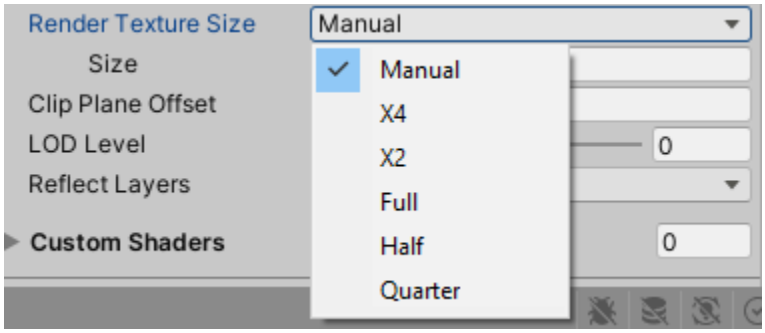
Filter mode is the created mirror texture (render texture) filter mode. So you can select the suitable mode according to your needs.

If checked disables the pixel (additional) lights for mirror render. So you can gain performance.

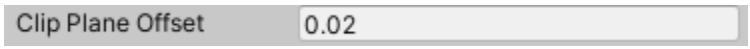
If checked, it enables the shadow on reflection render. But it may create performance costs!



It culls the reflection area in a spherical view. So you can create a max distance to draw the reflection. But it may create not drawn (long distance) areas. So it is good to disable that!

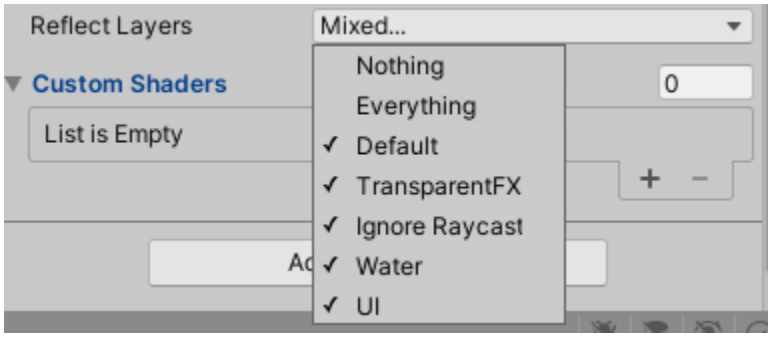


Render texture size is the size of the reflection. For perfect mirrors you should select the “Full” (it creates a screen sized reflection texture to fit exactly on the view). But you can select others or enter the “Manual” mode :)

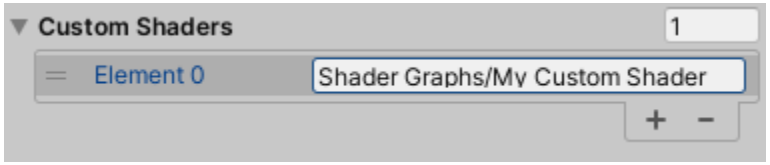


The clipping plane offset of the reflection from the game object. Making this “0” may create a little artifact. So it is good to hold it up a little bit.

Camera LOD Level to draw begin with. So if you have lods in your scene, you can start to render the reflection from a certain LOD level. So you may gain performance!



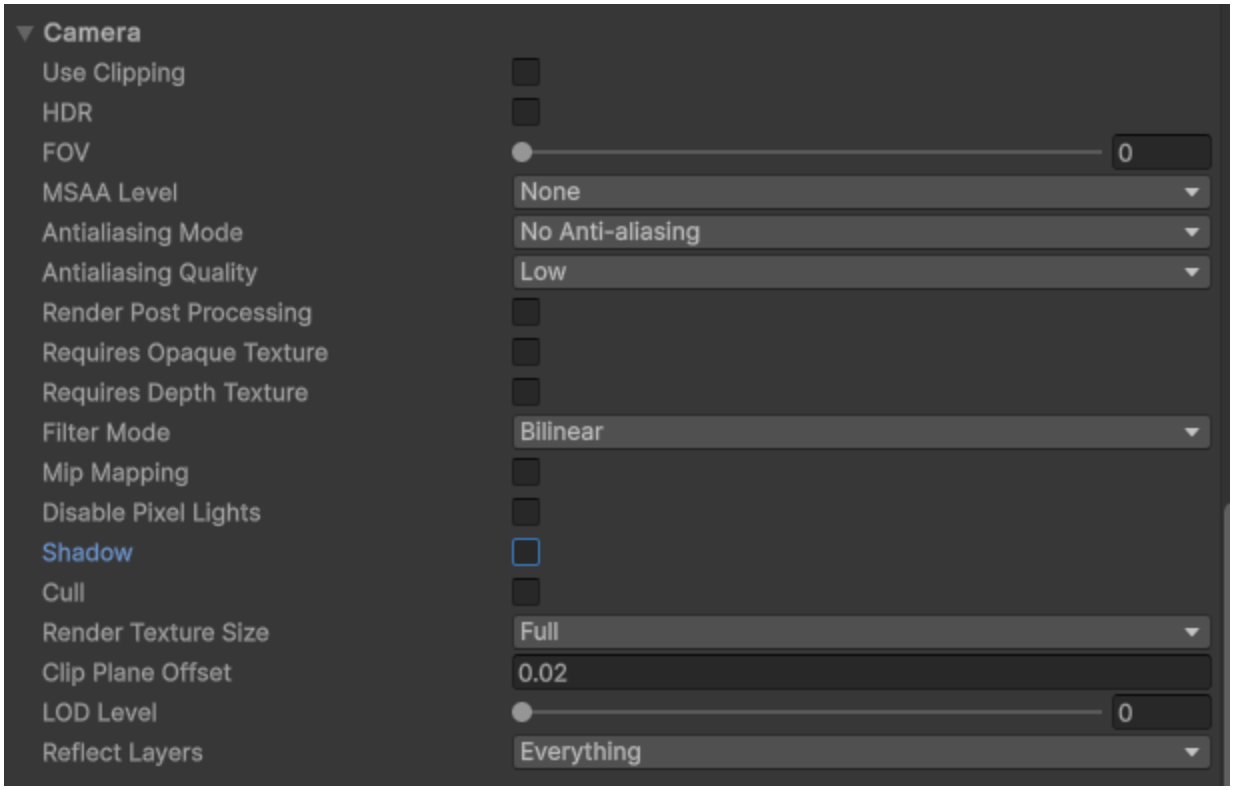
You can only reflect certain layers if you want. It is good to reflect everything but you may put your complex objects in a certain layer and exclude it here. So reflection can be drawn faster!



It will be easy to include your custom shader to be used with my reflection textures. This slot will be used for that purpose. Just add your custom shader string path here, and mirror manager will send the textures to your custom shader automatically :) Next version will hold an example for that!

Optimization “Mirror Manager Script -> Camera Settings”

Because the script draws the reflection as a second camera, the speed of the game may decrease dramatically. Try below actions to make it reasonable:



HDR: If you enable it, reflection will be drawn with HDR render texture. So it may decrease the speed a little bit.

MSAA Level: Enables the MSAA (Hardware anti aliasing) on reflection. So it may decrease the speed a lot especially on mobile platforms.

Antialiasing Mode: Enables the software anti aliasing mode on reflection camera. It may decrease the speed a little bit.

Render Post Processing: Enables the volume on the reflection camera. So post effects will be enabled on reflection and decrease the speed (like bloom).

Filter Mode: Enables the filtering on the reflection camera. So It may decrease the speed a little bit.

Mip Mapping: Enables the mipmaps on a reflection camera (calculates the mipmaps). Do not open it if you do not need it.

Disable Pixel Lights: As it says :) It does not draw pixel lights (spot and point). So checking it may give you extra speed.

Shadow: Enables the shadows on reflection. Do not check it if you do not really need the reflections.

Cull: It sets the culling distance for the camera to draw. So if you do not need to draw very distance, you can enable it and set a value for the distance.

Render Texture Size: Changes the reflection texture size. Smaller sizes are faster but may become blurry.

LOD Level: To use lower LOD levels to speed up the rendering (of course if you have LOD levels :))

Reflect Layers: Do not render very complex items. Just put them in a different LAYER and unselect that layer here :) So reflection camera will not render that layer (items).

Optimization Samples

Mirror script can drop your frame rate (especially on VR). Because it re-draws the scene from the mirror's object perspective (in VR it draws 2 times, one for each eye). So you have to be careful where to use it and how to use it!

For example:

1:

If you do not want a perfect mirror use the LODs to draw the reflection. For example in my car screens, that is a very detailed car and has 300K triangles. I am using a LOD for the car which has 10K triangles (LOD 1). Then I can select the “Camera LOD Level = 1” from the “CameraShade” script. So it forces the LOD1 (not the real car) for the reflection. So I gain a lot of performance but get smooth semi-reflections!

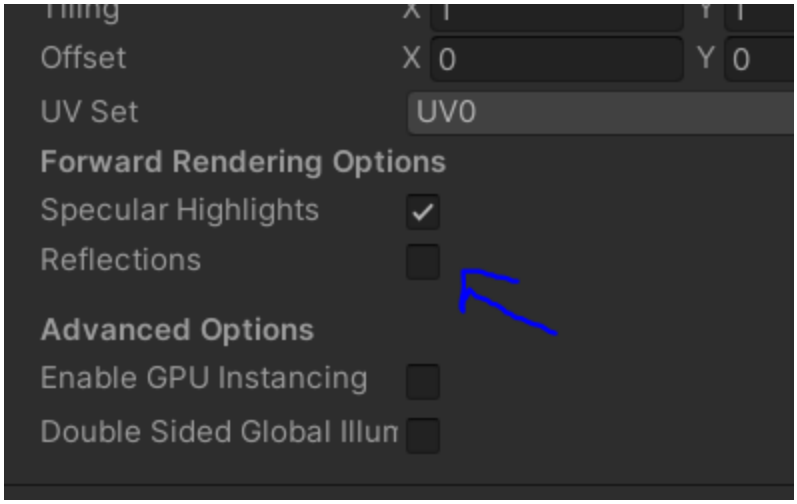
2:

Use different objects and shaders for reflections: You can put your reflection items to different layers and draw only them with the script's “Reflect Layer” option (your main camera also should not draw them again with its layering option). So you can use simplified objects (even with simple shaders) for reflection and gain a lot of performance.

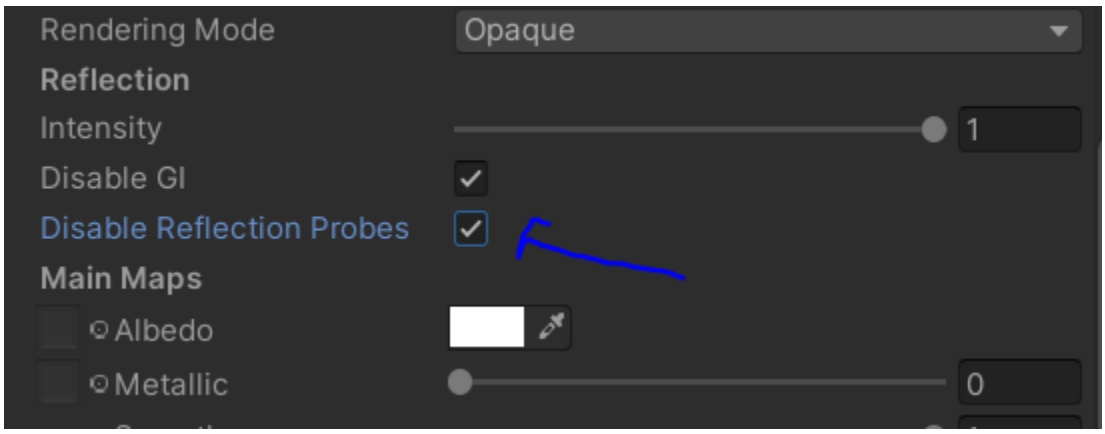
Troubleshoot:

Reflection probes mixes with mirror:

If you increase intensity of the reflection (smoothness), reflection probes may mix with the mirror texture and create unwanted results. So uncheck “Reflections” on shader for forward rendering path:

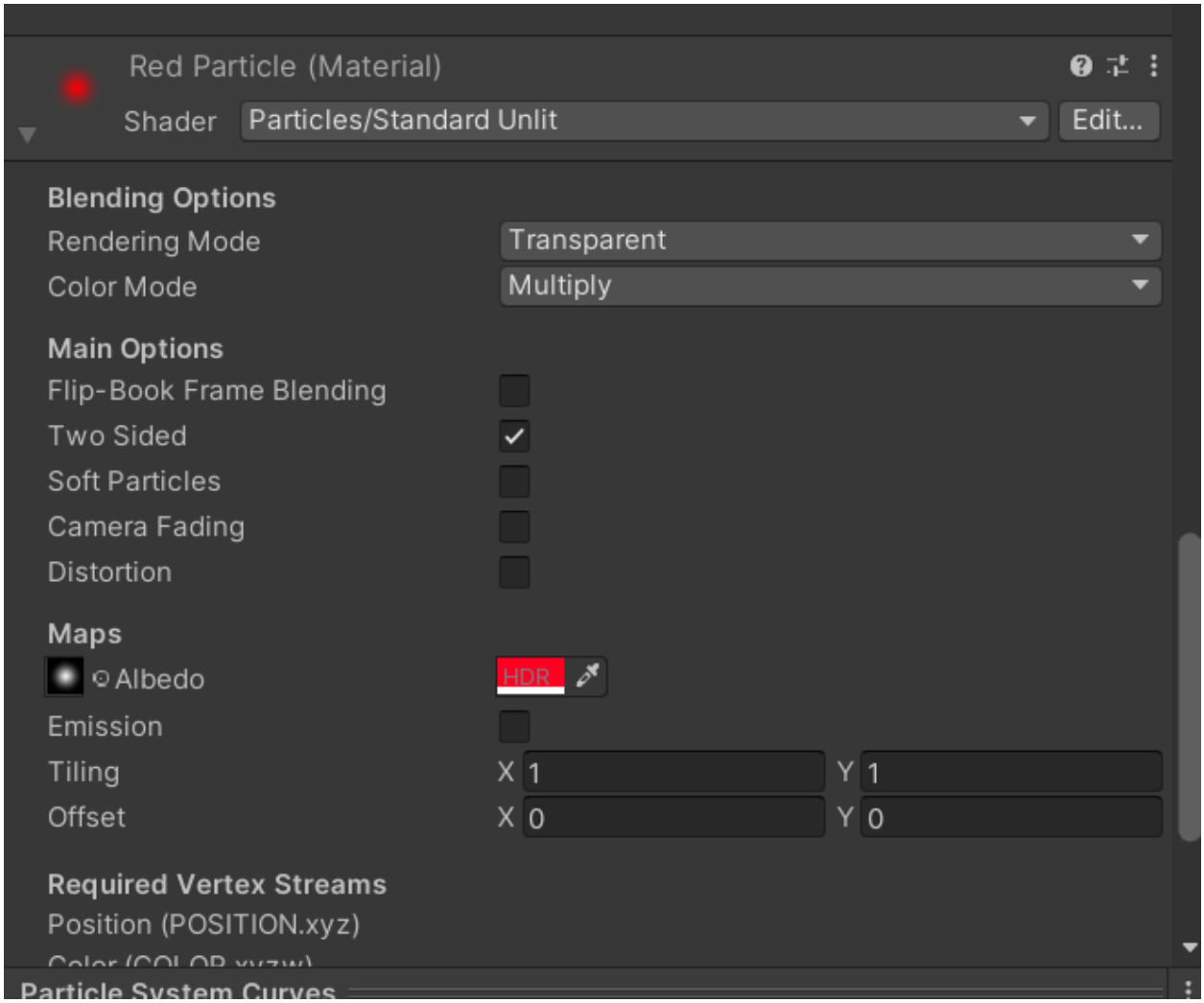


Or check the “Disable Reflection Probes” on shader on deferred rendering path:



Particles does not show:

Particles may not be seen in the mirror. The solution is actually easy. Just create a new particle material and mark “two sided - double sided” etc. Use that material on your particle and voila :) Image below shows the standard pipeline solution, but URP is the same too.


Red Particle (Material)
Shader
Particles/Standard Unlit
Edit...
Blending Options
Rendering Mode
Transparent
Color Mode
Multiply
Main Options
Flip-Book Frame Blending
Two Sided
Soft Particles
Camera Fading
Distortion
Maps
Albedo
HDR
Emission
Tiling
X 1
Y 1
Offset
X 0
Y 0
Required Vertex Streams
Position (POSITION.xyz)
Particle System Curves