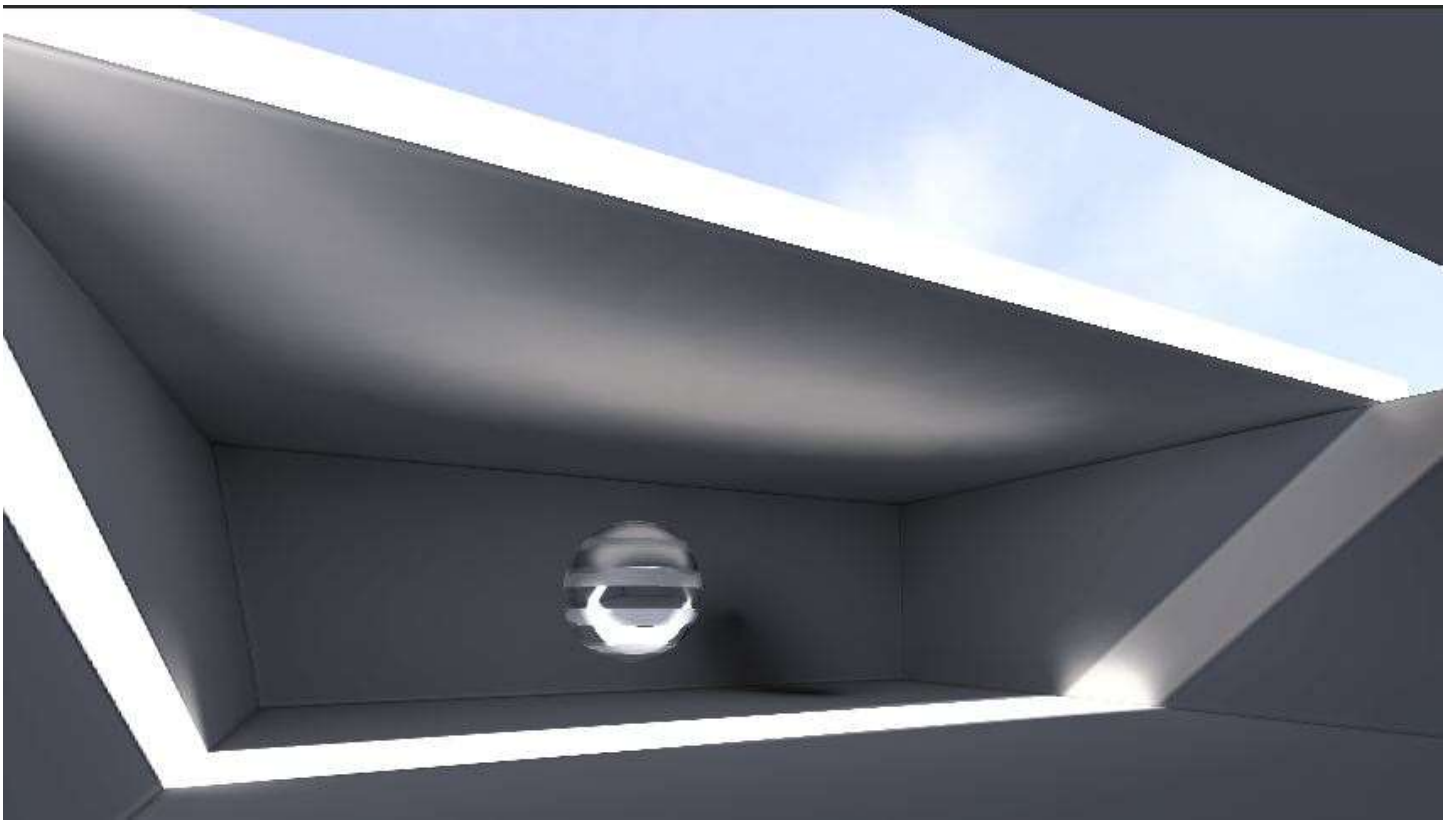
SkyCube RT Sample package

The sample scene show you how to use the tool.

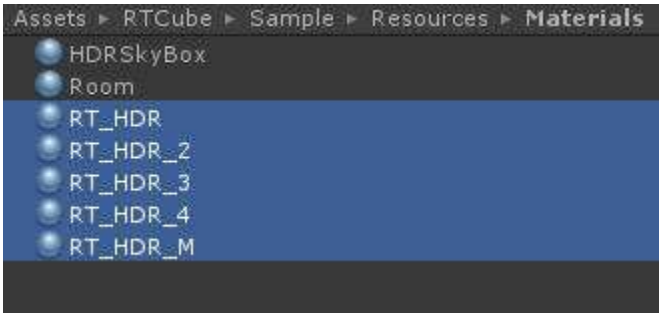


You can find HDR sample scene **Sample_rtref01.unity** under the path: \RTCube\Sample\

The sample come from the Demo: [HDR Real-time reflection with glossiness chain http://youtu.be/7GMmelHbB-U](http://youtu.be/7GMmelHbB-U)

Materials:

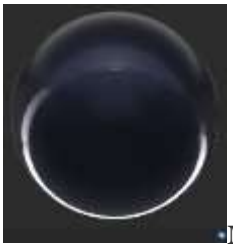
In this sample scene there are 5 sample materials to show you how to use it in your shader for multi roughness HDR reflection. You can apply them to moving ball to preview result by running it in editor.




Mirror


Mirror with different roughness


Different reflectance


Multi-Layer reflection (Plastic)

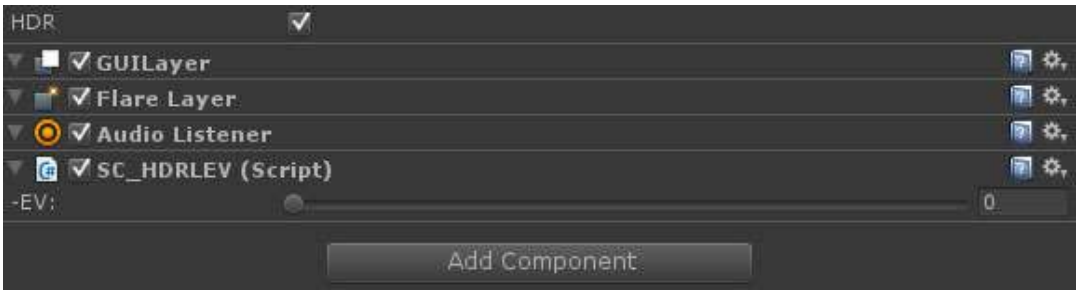

Multi-Layer reflection with color variation (Car Paint)

Camera:

The sample scene is HDR and over exposure, all of material use RGBM encoded HDR map as IBL map.

Before you check it, you better add an Exposure Value Control component: **HDR Rendering EV** to Main Camera and make sure the camera works in HDR mode.

Choose **Component-> Skycube-> Image Effects->HDR Rendering EV** from the menu bar (Its position was changed since v2.5):



The slider –EV just like the Exposure Value control in camera, but only affect scene intensity nothing about exposure. Since this is –EV, the number always negative in the case. The bigger number means the less brightness in HDR tone mapping. For instance, when the –EV=0 the camera will get 100% luminance from scene; when the –EV=1 the camera will only get 50% and so on. The following pictures show –EV from 0 to 8 in the sample.

For more details of Exposure Value please check:

http://en.wikipedia.org/wiki/Exposure_value

Shaders:

There are 5 sample shaders correspond 5 surface materials, you can find them under path: /RTCube/Sample/Shaders/

Basically, to use RGBM HDR cubemap to your own shader, you just need to copy **SC_Common_Libs.cginc** from Skycube package to your shader folder. And include this file under **CGPROGRAM** in your shader:

#include "SC_Common_Libs.cginc"

And make sure you work in D3D9 or higher. You can put following keyword to keep your shader compile for ShaderMode 3.0:

#pragma target 3.0

Then you can use the SC's lookup functions to repleace standard texture sampler functions, like tex2D(), texCUBE(). Please note, there are some SC lookup functions need keyword:

#pragma glsl

For more details of the functions inside **SC_Common_Libs.cginc** please check the shader Libs Reference under Doc folder.

Any question about this tool please contact me in mail: support@atomsdev.com