Planets Earth, Mars and Moon Shaders Asset

V0.4.4

Documentation

(Unity 5 version)

Charles Pérois - 2015

Table des matières

1. Introduction

This asset aims to provide a simple and fast way to add quite realistic looking planets Earth, Moon and Mars in your project. The newly released Unity 5 version features a 16K textured Earth.

The asset contains:

- A demo scene to show how to use it.
- Shaders for the Earth surface (16K, 8K and SM2.0 versions), clouds, clouds shadows, atmospheric outer scattering effect and Mars surface.
- Prefabs that can be dynamically instantiated.
- A high polygon sphere.
- A set of textures.
- A sunlight example.
- A Milky Way skybox.
- Scripts, mainly for the demo scene.

Unity 5 exclusive new features:

- Added a 16K textured version of the Earth (the 8k version still exists)
- All 8K shaders don’t use the 4 textures-tiling algorithm anymore, as Unity 5 raises the maximum texture resolution to 8192px, it’s no longer necessary and will saves some resources.

2. Release Notes

v0.4.4
- Rendering of the atmosphere has been improved: it no longer exaggeratedly bleeds over the dark side of the planets.
- 2 new parameters have been added on the atmosphere scripts to handle the above-mentioned bleeding effect and the strength of the scattering effect.
- Fixed the blue dark side bug on the 8K Earth.

v0.4.3
- New Earth night lights textures (a bit more photo-realistic).
- Color parameters for the night lights on the earth shader has been added.
- Fixed a bug in linear color space that made dark sides of both Earth and Mars blue.
- Added two new Earth and Mars prefabs optimized to render better in linear color space.
- Changed the clipping planes of the main camera of the demo scene to avoid Z-fighting artifacts.
- A “Quit” button has been added to the demo scene. (Yes it was about time)

v0.4.2
- Reworked Earth normal map. (both 8K & 16K)
- Fixed a bug with Earth specular map that caused black artifacts on the dark side of the Earth.
- Fixed a flickering/artifacts issue on the clouds shadows shader when using DirectX 11.
- Add a brightness parameter on the Earth ground shader.

v0.4.1
- Reworked Earth normal map. (both 8K & 16K)
- Fixed a bug with Earth specular map that caused black artifacts on the darkside of the Earth.
- Fixed a flickering/artifacts issues on the clouds shadows shader when using DirectX 11.
- Add a brightness parameter on the Earth ground shader.

v0.4
- ported to Unity 5
- Add a 16K textured Earth prefab
- Add 8K normal map for Mars
- Add a normal map strength parameter for Mars and Earth shaders
- Add a Luminosity parameter for cloud shader
- Some file renaming to prevent conflicts
- Add a “Hide GUI” option on the demo script
- Moon uses the Unity 5 Standard shader.
- Earth night lights tint a bit more yellowish.

v0.3.3
- Add normal mapping on the clouds shader.
- Add 8K texturing on earth normal map.
- Add 8K texturing on the earth night lights.
- Add 8K texturing on clouds color.
- Add 8K texturing on clouds normal map.
- New script for the demo scene.

v0.3.1
- Cloud “fake” shadows shader added.
- New Mars textures and normal map: now craters do look like craters, and hills like hills.

v0.3
- Mars and the Moon added.
- Planet prefabs now works "out of the box" using tag to get the light source, and can be dynamically instantiated.
- New demo scene in which you can browse the planets.
- Updated documentation.

v0.2.1
- New Earth surface texture.
- New Clouds Texture.
- Various tweaks on the atmospheric scattering default setting to make it look better "out of the box".
- Sun Lens Flare.

v0.2
- New earth surface shader that combines the 8k texture tiling and the inner scattering effect.
- Added Specular Mapping on the earth surface shader, with shininess, color and brightness setting.

v0.1
Initial release

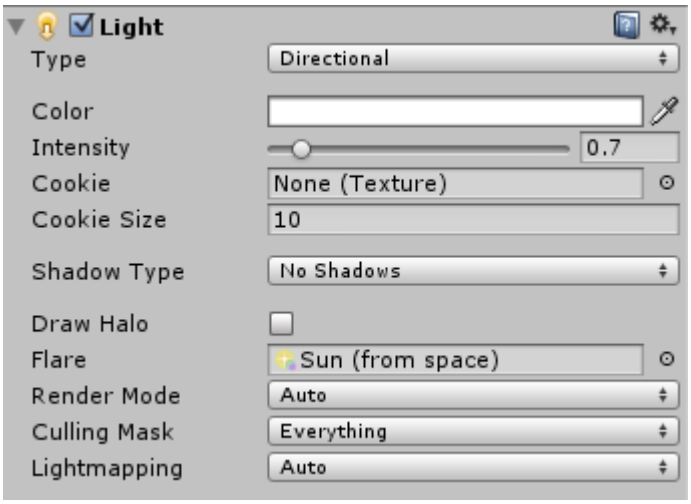
3. How to Use

1. Set the scene

First, copy all assets into your project.

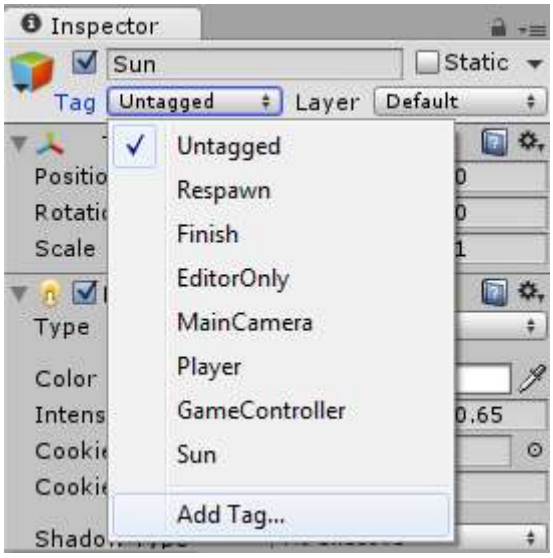
Then, if your scene doesn't contain any sun object yet (or any object that emit a light), create it.

Add a light to the Sun object (Add component > Rendering > Light), make it directional, then add a lens flare to it (one is included in "Lens Flares/" folder).

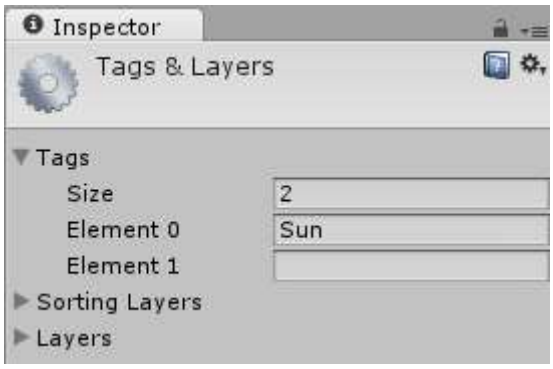


Finally, add a tag "Sun" to the sun object (or the one that emit the light). To can do it this way:

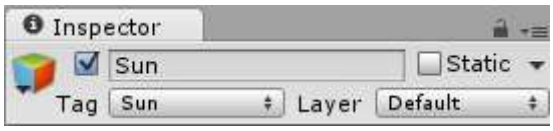
Select « Add Tag… »



Add a tag named "Sun"



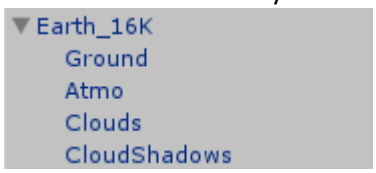
And select it in the drop down list



The tag have to be "Sun", this allows the planets prefabs to works without any setting, so that way you can dynamically instantiate them by code.

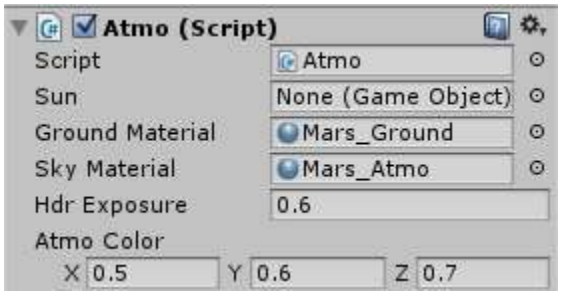
Now you just have to drag any of the planet prefabs to your scene. Those prefabs are contained in these folders:
Planet/Earth/Earth_16K.prefab
Planet/Earth/Eart_8K.prefab
Planet/Moon/Moon.prefab
Planet/Mars/Mars.prefab

Newly added prefabs should looks like this in the hierarchy view:



Prefabs can be dragged anywhere in the hierarchy, whether on the root or inside another object.

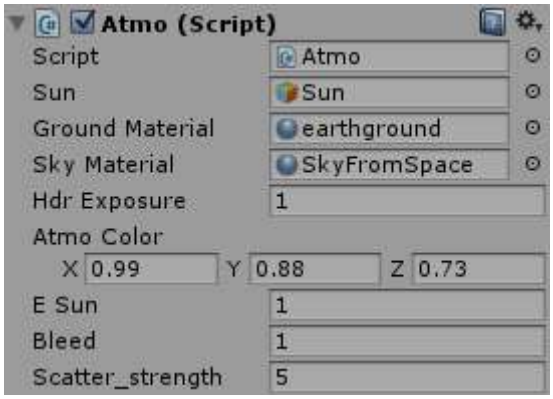
If for some reason your Sun object can't have a tag named "Sun" on it, you can still manually assign your Sun object to the planet object by filling the **Sun** field in the Atmo component in the Inspector (just drag the Sun object from the hierarchy view):



And that's it!

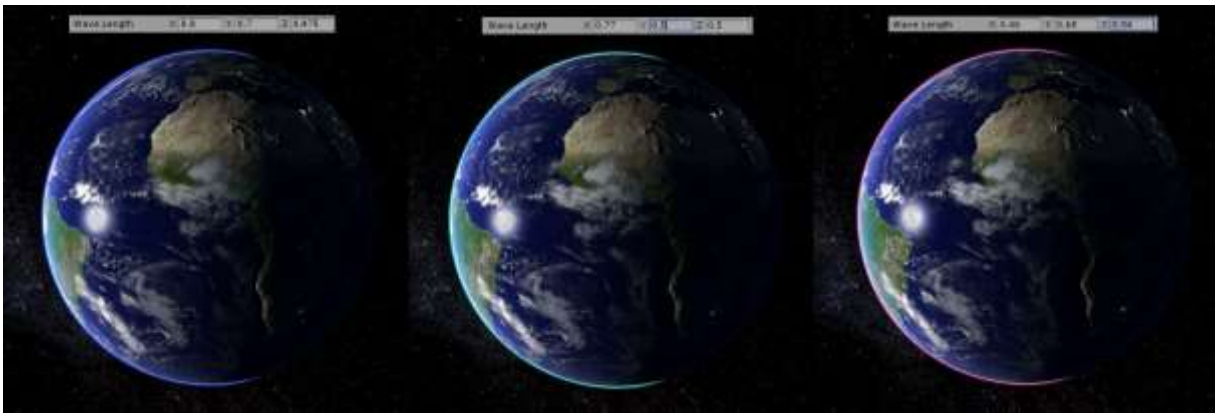
2. Outer atmospheric scattering effect

To customize the atmosphere rendering of Earth or Mars, (handled by both “Atmo.cs” script and “SkyFromSpace shader), select whether your planet object on the hierarchy view, or the prefab directly, and then go to the “Atmo (script)” section on the Inspector:



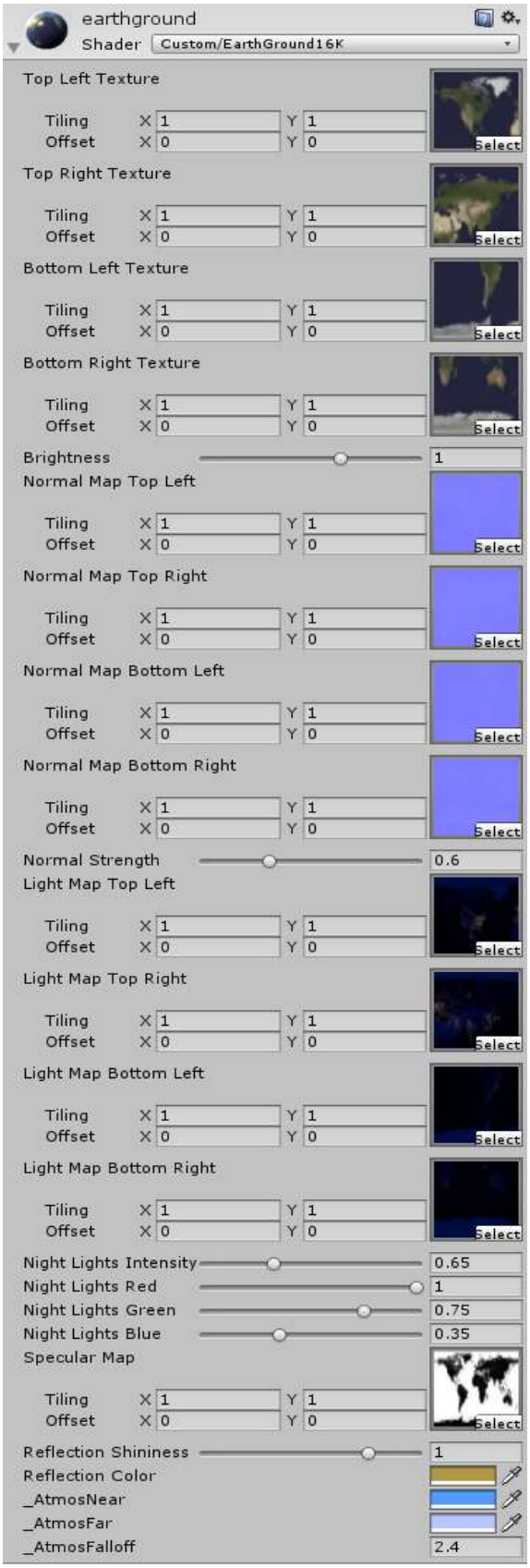
Sun: the sun object where the light is attached (not necessary if your sun has the “Sun” Tag)
Ground Material: the material that is used for the Planet object.
Sky Material: the material used for the Atmo object.
Both materials are located in Planets/<Planet Name>/Materials/.
Hdr Exposure: set the brightness of the atmosphere
The “*Atmo Color*” parameters plays with the wave length of the scattering algorithm, thus, play with will affect the color of the atmosphere.
Bleed: Set here how the atmosphere effect “bleeds” over the dark side of the planet. I made things so that “1” should give the best result, but in some condition you might want to reduce the bleeding of the atmosphere by increasing this value. (Please note that it could reduce the color gradient quality of the atmosphere if you increase it too much)
Scatter_strength: Hard to explain, but in short as you increase this value it will increases the width of the gradient color between the inner white and the end of the outer atmosphere. Eg on the earth, there will be more “blue”. Please note that as you increase this value, it will also increases the bleeding of the atmosphere over the dark side of the planet.

Be warned, the atmospheric effect doesn’t appear in the editor view (or barely appears but looks like crap), you have to run the scene to see it. So you should play with the value while the scene is running to see effects (remember that the changes you made while the scene was running will be lost once you stop it). Examples:



3. Surface materials shaders

1. earthGround16K



(Yeah that's a lot of textures.)

As you can see, most of textures are divided in 4 parts (except specular map). In this way we can have like an 8K texture on the sphere. Downward, you must attach your 4 texture parts like this:

Top Left Texture: put here the upper left part of your 16K texture
Top Right Texture: put here the upper right part of your 16K texture
Bottom Left Texture: put here the lower left part of your 16K texture
Bottom Right Texture: put here the lower right part of your 16K texture
Brightness: set here the brightness of the earth texture.

Normal Map: the normal map textures to add a relief (bump mapping) effect. If you want to use your own height map textures (the gray ones), be careful, as Unity, when it transforms them into normal map textures (aka tangent spaced textures), adds a 1 pixel smoothing edge on each texture that is clearly visible and not wanted here. So I would suggest to prepare your own tangent spaced textures (the pink ones) that do not contain this 1px edge before importing them into Unity.

Normal strength: Strength of the bump effect of the normal map. Type 0 here to deactivate the effect totally. 1 equals the default strength from the normal map texture.
Light Map: the illuminating texture that allow the city light to be shown on the dark side of the earth.
Night Light Intensity: set here the intensity of the night lights.
Night Light Red: set here the red value of the night lights color.
Night Light Green: set here the green value of the night lights color.
Night Light Blue: set here the blue value of the night lights color.
Specular Map: the specular map texture to set the sun reflection area (water here)
Reflection Shininess: set here the diffusion of the sun reflection on the water. Higher = less diffusion.
Reflection Color: set here the color of the reflection. By playing with the darkness of the color, you can play with the intensity of the reflection.

The next 3 parameters affect the atmospheric scattering gradient effect on the surface of the earth.

_AtmosNear: inner gradient color tint.

_AtmosFar: outer gradient color tint.
_AtmosFalloff: set here the thickness of the gradient. Lower value means a thinner atmosphering gradient ring.

If you want to use your own 16K textures, please note that you have to set their “wrap mode” on “clamp” (default is “repeat”)

2. earthGround

It’s the same as earthGround16K, but using single textures only (limited to 8K then)

3. earthGroundSM2.0

This shader is here to ensure a shader model 2.0 compatibility for directX9.0 GPU, as the 2 previous shaders are compatible with DirectX 9.0c GPU only.

It’s globally the same as earthGround, but without specular map (arbitrary choice, I think the specular mapping is less important as other effects).

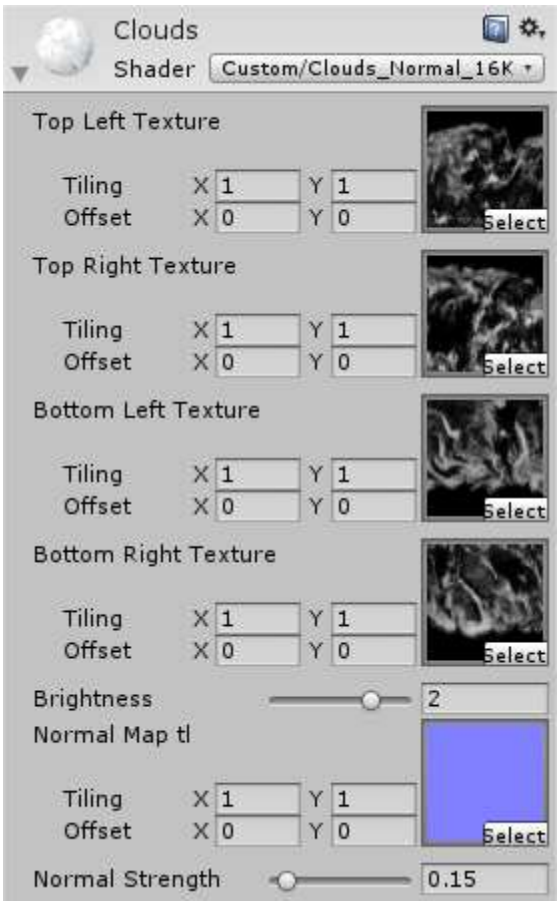
Please also note that if a directX9.0 or lower GPU is used, you will also lose the outer atmospheric scattering effect.

4. surface_scattering

This shader used for the Mars surface, is basically a simplified version the “earthGround” shader,it loses the nightlights and specular effects.

4. Clouds and cloud shadows material shader

1. Clouds_normals_16K



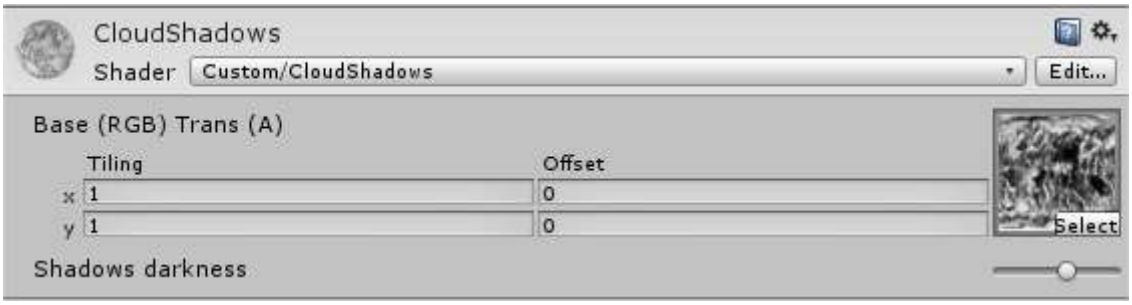
Top Left Texture: put here the upper left part of your 16K texture
Top Right Texture: put here the upper right part of your 16K texture
Bottom Left Texture: put here the lower left part of your 16K texture
Bottom Right Texture: put here the lower right part of your 16K texture

Brightness: The brightness of the clouds.
Normal Map: the normal map texture to add a relief (bump mapping) effect.
Normal strength: Strenght of the bump effect of the normal map. Type 0 here to deactivate the effect totally. 1 equals the default strength from the normal map texture.

2. Clouds

A simplified version of the cloud_normals_16K shader that uses a single texture.

3. Cloud Shadows



The texture here should be more or less the same as the clouds one.

"*Shadows darkness*" setting here changes the opacity of the shadows from 0 to 1. Default is 0.5.

In both case just be sure that the texture have "alpha from grayscale" checked.

5. Other Scripts

There are some scripts I haven't talk about in this documentation, here's theirs purposes:

- CameraZoom.cs: Allow to zoom-in and zoom-out usin the scroll wheel. Attach it to the camera.
- EarthMovement.cs: Allow any object to automatically rotate. You can attach another object to be synchronized to the one you attached the script on, like it's used for the clouds and its shadows.
- RotateEarth.cs: Allow users to rotate a planet with the right mouse button.. Attach it to any planet object.
- RotateSun.cs: Allow users to rotate the sun light with the left mouse button or the Q-D keys. Attach it to the "Sun" object.
- RotateCamera.cs: Controls most of the camera "flyby" commands.
- MainGUI.cs: This script allow to browse the different planets using buttons in the demo scene. You can see in it how to dynamically instantiate the planets.

4. Credits

Cloud and ground shaders are based on the work of Fragile Earth Studios:

http://fragileearthstudios.com/2012/07/03/code-snippets-realistic-earth-and-clouds-shader/

The atmospheric scatter shader is a part of “scrawk” work that you can find here:

http://scrawkblog.com/2013/04/13/gpu-gems-to-unity-atmospheric-scattering/

Earth textures came from NASA, have been modified to render better within the scene, but you’ll find originals here in the Blue Marble collection:

http://visibleearth.nasa.gov/view_cat.php?categoryID=1484&p=1

Examples:

Earth surface : http://visibleearth.nasa.gov/view.php?id=57752

Night lights : http://visibleearth.nasa.gov/view.php?id=55167

Clouds : http://visibleearth.nasa.gov/view.php?id=57747

Normal map : http://visibleearth.nasa.gov/view.php?id=73934

Specular map : http://www.shadedrelief.com/natural3/pages/extra.html

Moon and Mars texture can be found on the Celestia Motherlode website:

Mars textures: http://www.celestiamotherlode.net/catalog/mars.php

Moon textures: http://www.celestiamotherlode.net/catalog/moon.php

Skymap textures uses a photograph from the European Southern Observatory (ESO) / S. Brunier :

http://commons.wikimedia.org/wiki/File:ESO_-_The_Milky_Way_panorama_(by).jpg