PLANET SHADER AND SHADOWING SYSTEM

UNITY 5

Quentin Muntadas

2015

Contents

Introduction

This asset is a project made to help you have realistic looking planet and provide you with an easy and cheap way to produce high quality planetary shadow.

This package also come with mobile version of the shader.

This asset contains :

- Prefab of the earth and an alien planet with a ring.
- Two demo scene.
- One shader for the planet and one for ring. Works with a script to make it work on mobile or remove the atmosphere effect.
- 4 models of highpoly sphere (standard sphere and isocahedric sphere).
- Set of high-res texture for both planet.
- An high-res milky way skybox.
- A sun lens flare.
- A set of script to handle the shadow, movement, rotation...

Version history

v1.0 – Orignal Release.

Setup

Quick guide to put a planet in a scene.

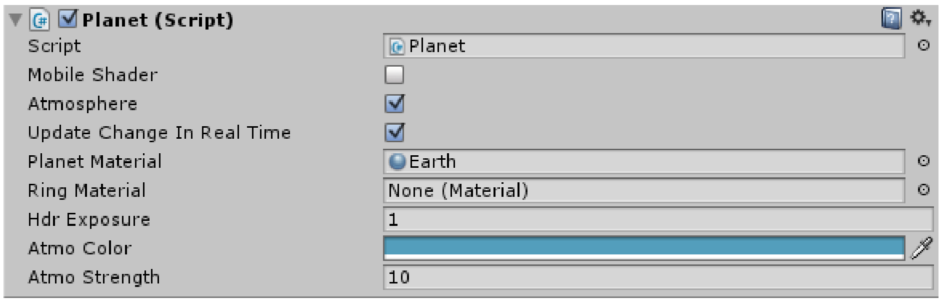
1 : Make sure that a directionnal light is present in the scene.

2 : Take a prefab from PlanetShader -> Prefab -> planet and put it in your scene.

3 : Voila ! Your planet should be showing on your screen !

Basic Introduction to the planet script

The planet script should be attached to every planet.



Planet (Script)

Script	Planet
Mobile Shader	☐
Atmosphere	✓
Update Change In Real Time	✓
Planet Material	Earth
Ring Material	None (Material)
Hdr Exposure	1
Atmo Color	🟦
Atmo Strength	10

/!\ Caution : Two planet shouldn't share the same Material, make sure you made a custom material for each planet you have in your scene !

Mobile Shader : Choosing this option will change how the normal map are handled in the shader (because mobile device handle normal map differently). This will also remove the shadow (for better performance).

Atmosphere : Choosing this option will tell the shader to calculate and display the atmosphere of the planet.

Update Change In Real Time : Choose this option if your planet is moving or if you modify properties of the planet during game time, otherwise I suggest you uncheck it for more performance.

Planet Material : (shader : Planet/Planet)
The material of your planet.

Ring Material: (shader : Planet/PlanetRings)
The material of the ring of your planet. Puting it here will make the planet automatically project a shadow on it (without the help of a script).

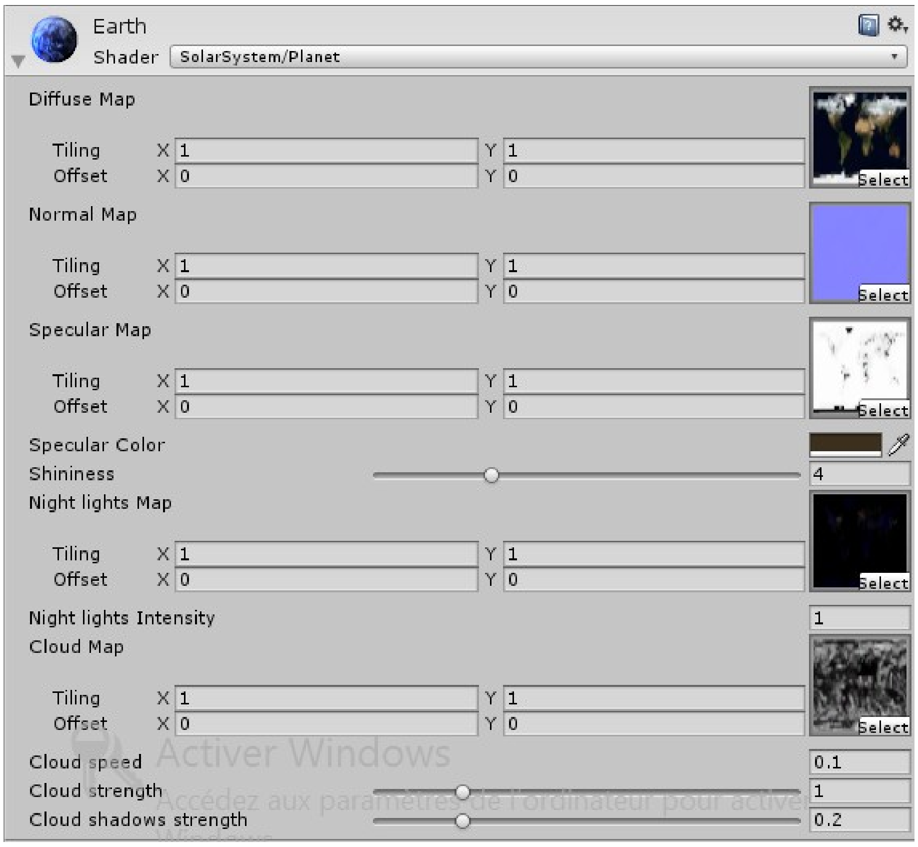
HdrExposure : Control the hdr exposure of the planet.

AtmoColor : Color of the atmosphere.
Make it all black for planet without atmosphere.

AtmoStrength : Strength/Opacity of the atmosphere.
This option will change both the strength and size of the atmosphere.

Material / Shader

Planet Ground Shader :



```
Earth
Shader  SolarSystem/Planet
Diffuse Map
  Tiling   X 1   Y 1
  Offset   X 0   Y 0
Normal Map
  Tiling   X 1   Y 1
  Offset   X 0   Y 0
Specular Map
  Tiling   X 1   Y 1
  Offset   X 0   Y 0
Specular Color
Shininess                 4
Night lights Map
  Tiling   X 1   Y 1
  Offset   X 0   Y 0
Night lights Intensity    1
Cloud Map
  Tiling   X 1   Y 1
  Offset   X 0   Y 0
Cloud speed               0.1
Cloud strength            1
Cloud shadows strength    0.2
```

Diffuse : This is the diffuse map.

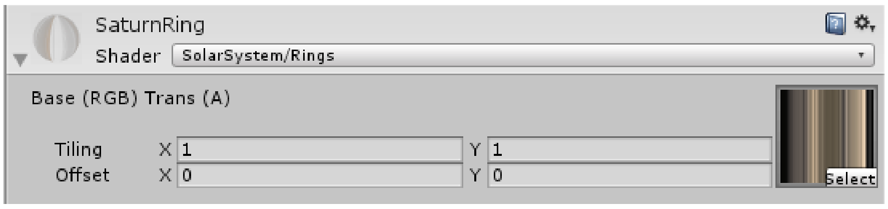
Normal : Bump map.

Specular : Specular map. You can adjust the size of the specular point and his color.

Night lights : Diffuse map for the dark side of a planet. You can adjust the intensity of it. You may want to have a brighter value if you are working in a gamma color space.

Cloud : /!\ MAKE SURE THAT YOUR CLOUD TEXTURE WRAP MODE IS SET ON REPEAT.
Diffuse map for the cloud of the planet. They will be rendered above the other texture. They will also hide part of the bump map where the cloud are strong. You can change the strength of the cloud and can also add a shadow effect under them with the "cloud shadow strength" slider.

Rings shader :


SaturnRing
Shader SolarSystem/Rings
Base (RGB) Trans (A)
Tiling X 1 Y 1
Offset X 0 Y 0

This shader is a port of a the Unity Unlit transparent shader expect this shader handle our sphere shadowing.
This shader only accept a diffuse map. Just make sure that this map is transparent !

Lighting

These planets must be rendered with a directionnal light as the shader is computed "as if" the main light was a directionnal light.

The planet will react accordingly to the main light intensity and color.
The planet are not sensible to fog.

Shadow

You can add shadow to your planet to make an eclipse-like effect.
The demo-scene as a shadowManager script that will give all the planet the coordinates of the other planet to compute the shadows.

Night light of planet are affected by those shadow.
The shadow is computed with the planet coordinates and their scale. Then the shader compute the shadow with a simple sphere formula.
This make this shadow system really performant and cheap (works on mobile), and is also of a way better quality than the unity built-in shadow system.
Unfortunately, because of progamming language limitation, the limit of planet shadowing each other is limited to 10. If you really need to have more shadow, feel free to email me and i will tell you a way to go past trough this limitation.

Model

This package come with 5 models, of which 4 are sphere model.
The other one is the ring model for saturn.

The sphere model are of two types, basic sphere mapping and isocahedron mapping. Isocahedron are better looking and more performant but some mapping error can appear at the pole sometimes.

Color space

The planet script automatically adjust the planet color accordingly to your color space. This way, planets are always rendered in linear color space, even on mobile when linear color space is not supported yet.

Scripting with the shadow system

This chapter is for people who want to know a little about the shadow system and how to interact with it.
I advise to read this only if you know a little about programing first.

To add shadow to your planets, you have to give to the planet of your choice the coordinates of the planet who will project shadow into it.
For example, if I want an eclipse-like effect on the earth, i will have to give the moon coordinates to planet script of my earth. But this will only project shadow on earth, so the moon wont be "shadowed" by the earth. To do this, simply give the moon the earth coordinates !

For advanced user, here are the core function (setter) :

void setShadowNumber(int n);

This will set a limit (n) for how many shadow your sphere will compute (maximum 10).

void setShadow(Transform p, int n);

This will assign the Transform p to the position n in the coordinates array.
n must be between 0 and 9.

In actual code, to give a planet another planet coordinates, you will to do it like this in c# :

```
GameObject myPlanet; // create a GameObject called myPlanet

myPlanet = GameObject.Instantiate(myPlanetPrefab, Vector3.zero, Quaternion.identity) as GameObject;
// create a planet with null position and rotation and assign it to myPlanet.

myPlanet.GetComponent<Planet>().setShadowNumber(1);
// This line will tell the GameObject myPlanet that it will have to compute a given number of shadow,
// here only one. MyPrefabPlanet must have a planet script attached to it otherwise unity will return an error.

myPlanet.GetComponent<Planet>().setShadow(anotherPlanet.transform, 0);
// This line will tell the gameObject myPlanet that i want it to compute the shadow of anotherPlanet. The second
// argument of setShadow is 0 because my array is only one element wide in this example and the first element of
// an array is in the position 0.
```

Don't hesitate to take a look at the ShadowManager.cs script to see how to completely shadow your scene.

About mobile rendering

Choosing the mobile shader option in your planet script will make your planet render with a modified version of the planet shader. It's still the same file, and the quality of rendering wont be altered.

Only difference between mobile shader and "classic" shader are :

- Normal map aren't handled the same way.
- Shadow isn't supported on mobile.

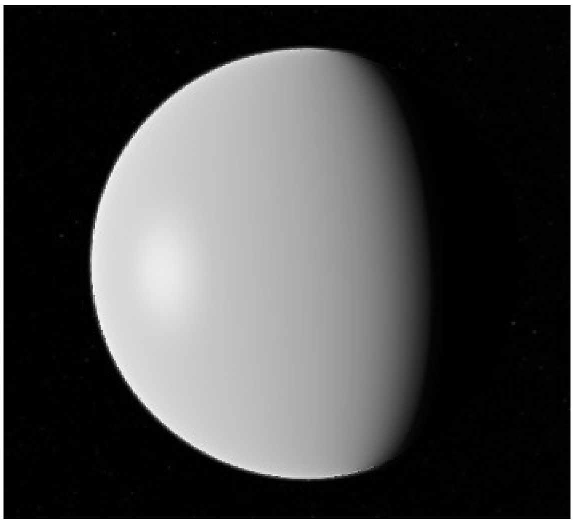
When building your project on mobile, you should be aware that mobile don't really like high-resolution texture. 8K texture is a bit too much so I suggest lowering your texture to 2k so that more phone could run your app (8k texture aren't that usefull on mobile screen anyway).

Making your own planet

Here are the step to make your own earth from scratch !

1 : Make sure that a directionnal light is present in the scene.

2 : Go to PlanetShader/Prefab/Planet/TemplatePlanet and drag it in your scene.


The template planet

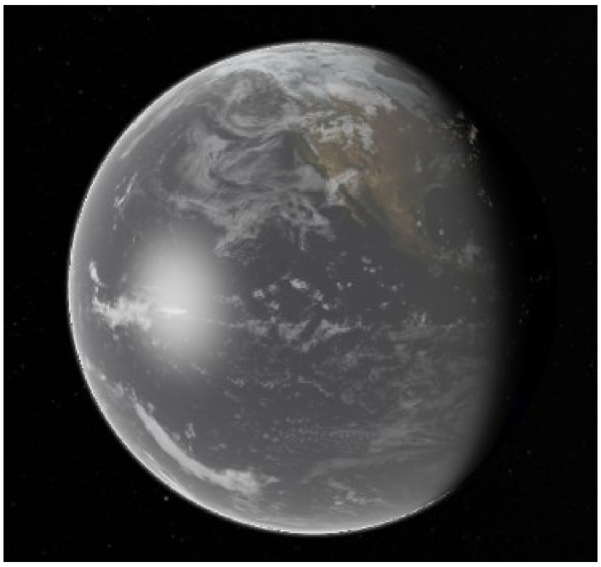
3 : Create a custom material in your project folder who will have the shader Planet/Planet.

4 : Assign the ground material to the mesh renderer component of your planet and to the "Ground material" field of your planet script.

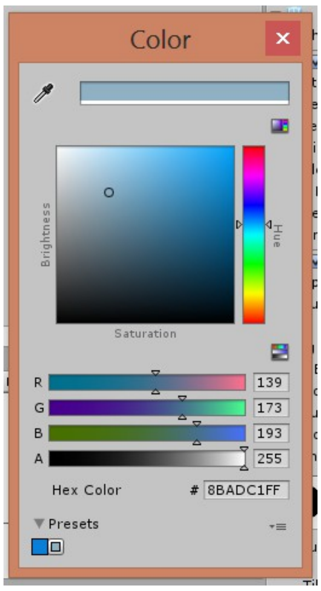
5 : Adding texture to your shader : For our texture, we will simply took the texture in the folder PlanetShader/Texture/Earth. Then it's pretty simple, expand the myPlanetGroundMaterial in the inspector and add the corresponding texture to the correct field.

EarthDiffuse for the diffuse map.
EarthNormal for the bump map.
EarthSpecular for the specular map.
EarthNight for the night light map.
EarthCloud for the cloud map.

This should be your result :

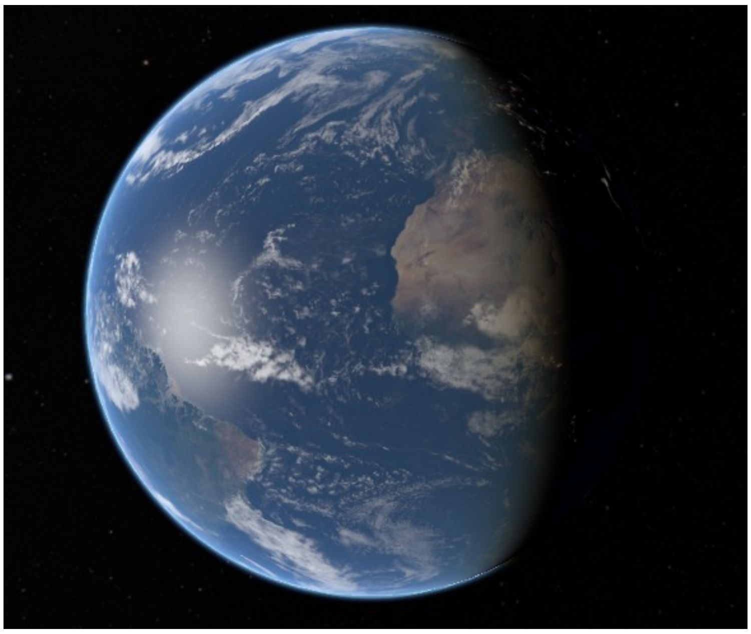


10 : To give the earth a better tint, go to your planet script and modify the value of the field Atmo Color. Click on the black bar and choose a color a bit closer to the earth atmosphere.
Like this one for example !



R 139
G 173
B 193
A 255
Hex Color # 8BADC1FF

Giving us this :



Your earth is now finished, you can still tweak the value in the planet script or in the earth ground material to make it look more like the real one.
For example, you could change the atmosphere strength or change the specular color.
Have fun !

Credit

Earth textures are available here :
http://visibleearth.nasa.gov/view_cat.php?categoryID=1484

Atmospheric scattering shading :
http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter16.html

Milkyway skybox :
http://svs.gsfc.nasa.gov/cgi-bin/details.cgi?aid=3895

Alien planet texture (New Aruba) :
http://freebitmaps.blogspot.fr/2010/11/srgb-planet-blink.html

Contact

If you have any trouble with this package, feel free to contact me to :
quentin.muntadas@orange.fr