SkyMagic - Guide

MacGruber's Tutorial Series

This tutorial gives you a quick introduction on how to use the SkyMagic plugin.

Overview

Introduction

SkyMagic is a plugin that allows you to bake and import custom global illumination into VaM. It can be used in a number of different ways:

- Bake custom reflections within VaM to have global illumination match your light setup perfectly.
- Use the baked ReflectionProbe of a scene/prefab imported from Unity as global illumination.
- Use some nice a 360-panorama, for example from hdrihaven.com, as global illumination.
- A combination of these things.

To make this possible SkyMagic consists of 3 parts that work together:

- **SkyMagic Exporter**: VaM plugin that takes a cube-screenshots of your VaM scene and saves as *SMC* file.
- **SkyMagic Importer**: Unity plugin that allows Unity to import the *SMC* file and automatically produce a cubemap texture and a ReflectionProbe prefab out of it.
- **SkyMagic Loader**: VaM plugin that retrieves a cubemap texture from a ReflectionProbe inside your CustomUnityAsset and sets it as source for VaM's global illumination. This is what you need to include when you release your scene.

The two VaM plugins are part of the *"MacGruber.Essentials"* plugin collection, so you have an easy VAR file to reference and many people will have it anyway.

The Unity plugin needs to be installed manually into your Unity project. My earlier tutorial *"Unity AssetBundles for VaM 1.xx"* is highly recommended, if you are not familiar with the process how to generate AssetBundles for VaM.

SkyMagic Exporter

To take a cube-screenshot within VaM, you need to place the *SkyMagic Exporter* plugin in your scene. I recommend to put it either on an *"Empty"* atom or the *"CustomUnityAsset"* you will later use to load the ReflectionProbe. You can find the plugin in the *MacGruber.Essentials* VAR package.
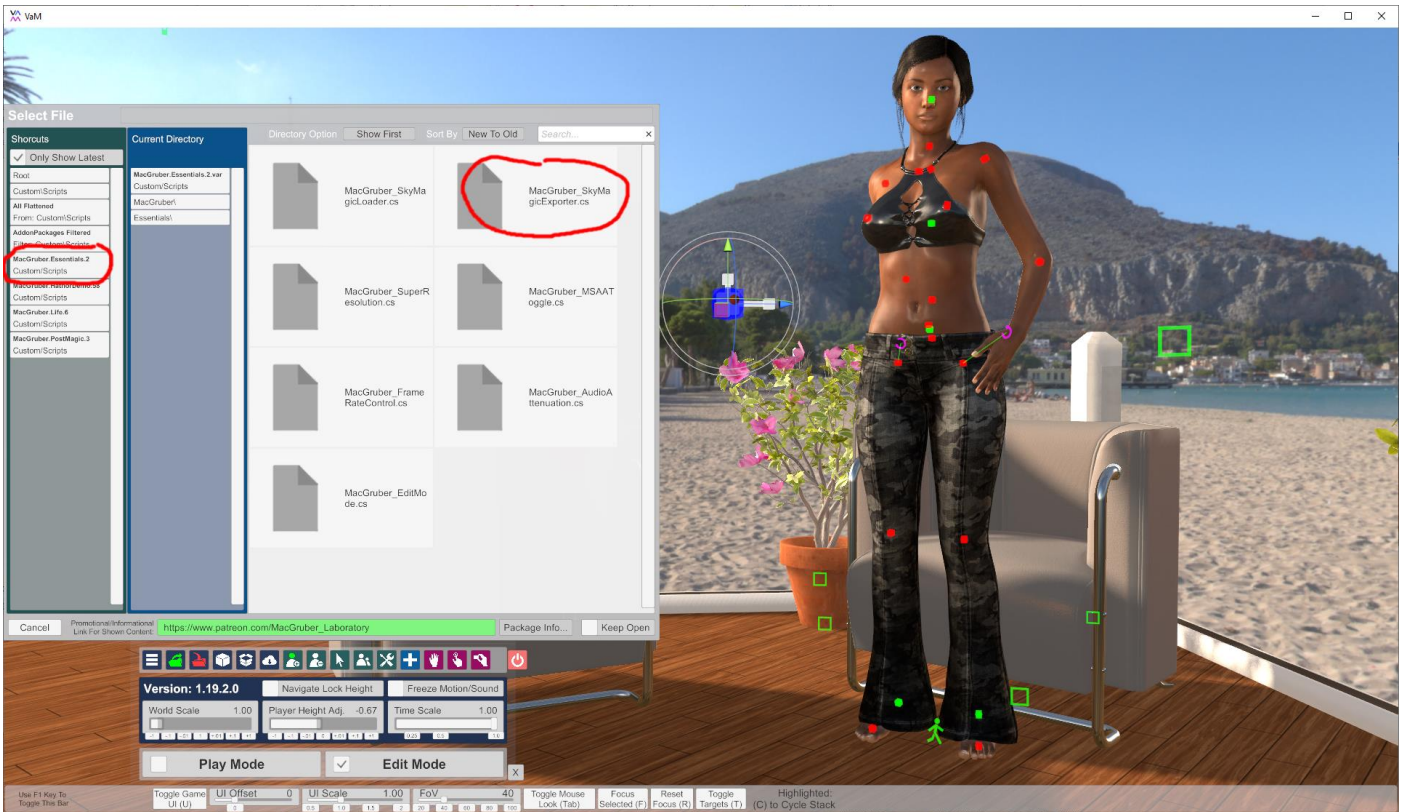
However, there are a few things to consider:

1. The position/rotation of your atom are important, as that is the position/rotation the plugin uses to take the screenshot. You will likely want it somewhere in the center of your scene.
2. In most cases it will be useful to temporarily disable characters, nearby smaller objects and everything that might move before taking the screenshot. The reason is that you take a screenshot from a single position for the entire scene. You can easily disable objects without removing them by using the *"On"* checkbox in the control tab of any atom.
3. You might also want to temporarily set the *"Global Illum Master Intensity"* slider in the Scene Lighting settings to zero.
4. A resolution of 256 or 512 will be enough in most cases, as your reflections will usually be rather blurry anyway. However, you can go up to 1024, if needed. With the cubemap format used by SkyMagic, anything larger would cause the texture size to exceed 8192.
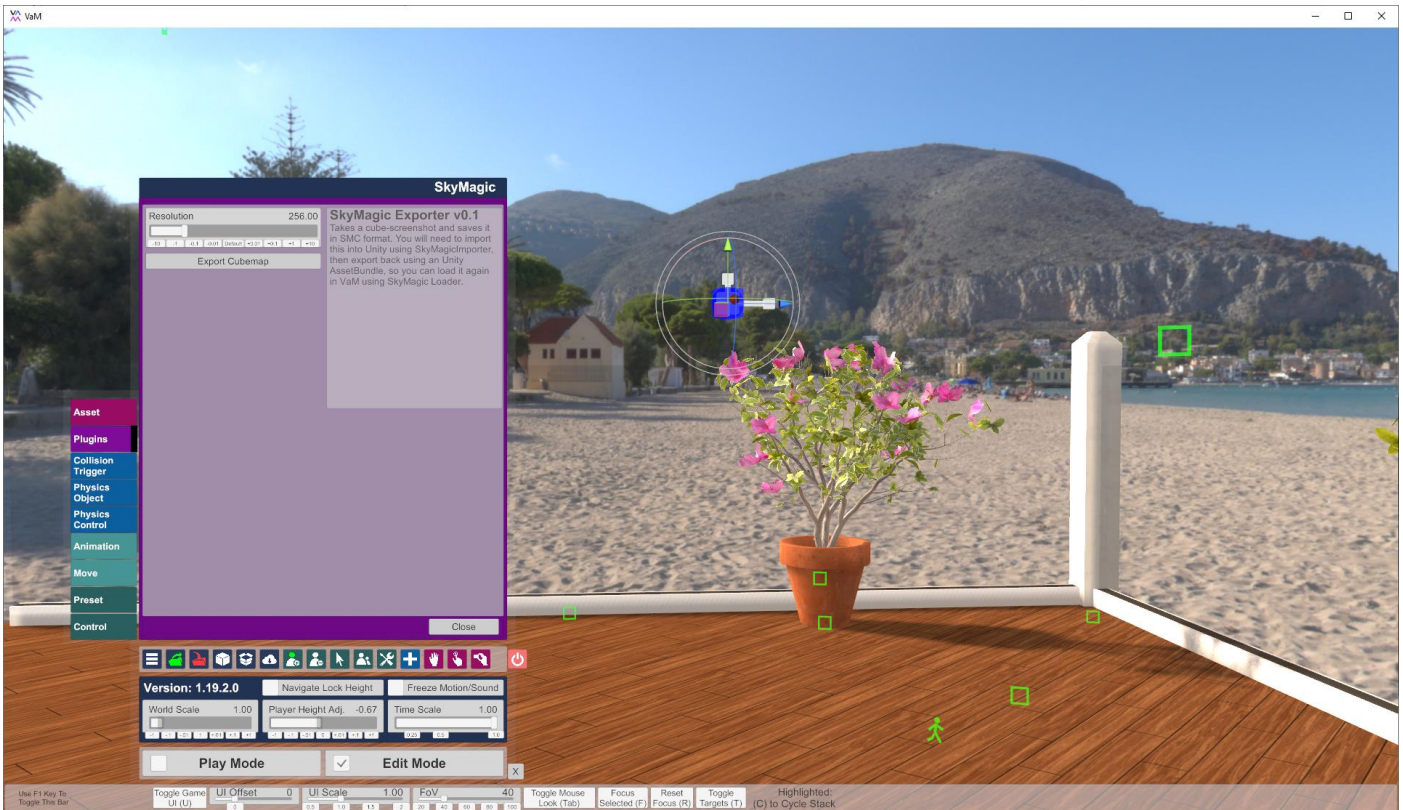
Otherwise the plugin is easy to use, essentially you press "*Export Cubemap*" and you get a file dialog where to save the file. By default, the *SMC* file will be put into *Saves\screenshots*. VaM's security restrictions won't allow you to save outside the VaM folders.

Ok, what's an *SMC* file, you never heard of it? This is my fancy invention, it stands for "*SkyMagicCube*" and it is nothing else than a plain *PNG* file containing the 6 sides of the cubemap. However, the file is Base64 encoded, simply because as of VaM 1.19.2 plugins are not able to save binary files directly. Yay! But...this is where *SkyMagicImporter* comes in, it can read the Base64 encoding and write out the contained *PNG* file.

SkyMagic Importer

This is NOT a VaM plugin, you need to put this **inside your Unity project**. For example, you can use the ready-to-go project from the *"Unity AssetBundles"* tutorial. You can put the *SkyMagicImporter.cs* anywhere in your projects Assets folder, but make sure it is somehow inside a folder name named "Editor", otherwise it won't compile.

Importing of *SMC* files generated by *SkyMagicExporter* happens automatically. You only need to copy the *SMC* file somewhere in your Unity project's *Assets* folder. The importer will detect and process it, which can make Unity freeze for a few seconds. The output of the plugin are two files:

- The extracted *PNG* file, with the texture import settings already set correctly. See left picture below. Should you want to use your own cubemaps from other sources, the important settings are:
  - Texture Shape: Cube
  - Mapping: 6 Frames Layout (Cubic Environment)
  - Convolution Type: Specular (Glossy Reflection)
  - Fixup Edge Seams: Enabled
- A prefab containing a ReflectionProbe, with above cubemap already linked. See right picture below. This is what you need to export as AssetBundle.

Details on exporting things from Unity can be found in my *"Unity AssetBundles"* tutorial.
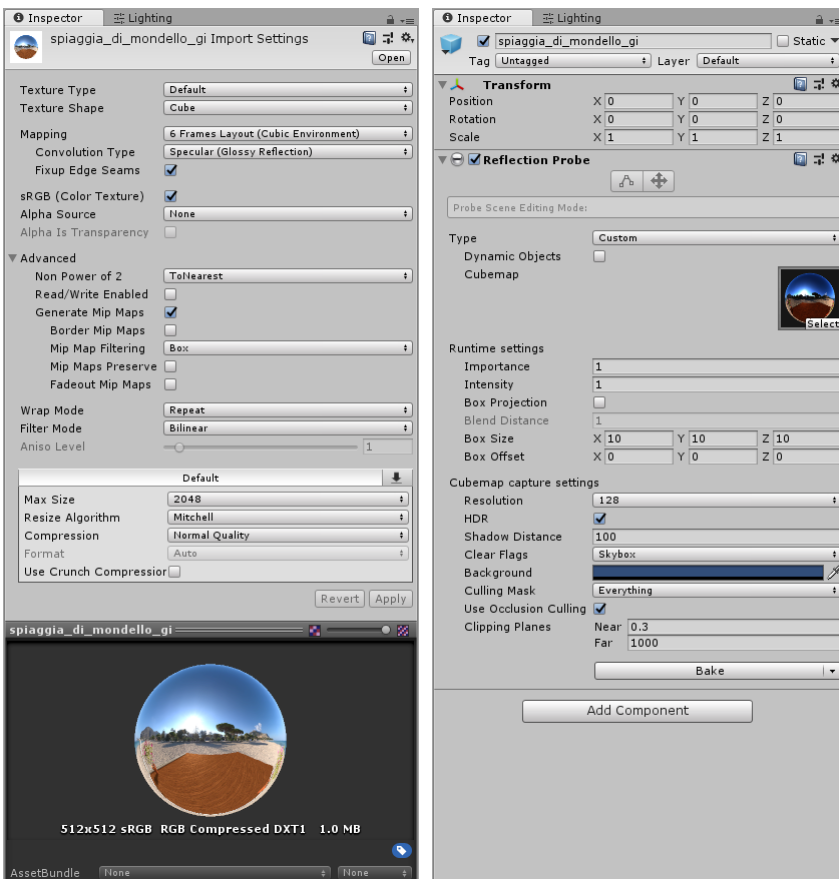
SkyMagic Loader

The prefabs or scenes containing your data need to be loaded as usual with a *CustomUnityAsset* atom.

Place the *SkyMagicLoader* plugin on the same atom. Enable the *"Global Sky Enabled"* checkbox to use the *ReflectionProbe* found in this *CustomUnityAsset* as source for global illumination.

You can now check your *"Scene Lighting"* settings. If your results are somehow way to bright or way to dark, you might want to try if changing the *"Global Sky is sRGB"* setting back in the *SkyMagicLoader* plugin helps.

While you can enable *"Show Skybox"* to use the loaded cubemap not just as source for global illumination, but also as skybox, it will have rather low resolution. You should use this mostly for testing, to make sure you got the right image. See the section on *"High Resolution Skies"* for more details how to import a sky like the one used in the demo scene properly.


The *SkyMagicLoader* plugin has a number of additional features you might find interesting:

- The prefab you load out of the AssetBundle can contain more than just a *ReflectionProbe*. For example, the *ReflectionProbe* could just be part of an entire scene you import from Unity.
- The prefab can contain multiple *ReflectionProbes*, if you like. You can use them to provide more detailed reflections for Unity native assets. See Unity documentation on *ReflectionProbes*. However, you need to make sure you set the *ReflectionProbe* type to *"Custom"*. Because of limitations of VaM, for VaM native assets only a single source for global illumination is supported. The plugin will use the first *ReflectionProbe* found in you prefab.
- You can use the *"ReflectionProbe Intensity"* slider to set the intensity of all *ReflectionProbes* inside your prefab. Use this, together with the *"Scene Lighting"* settings, to balance VaM global illumination vs. Unity global illumination.
- You can have multiple *SkyMagicLoader* plugins in the same scene, as long as only one of them has *"Global Sky Enabled"* checked. This makes sense if you want more *ReflectionProbes* for your Unity native assets and want to use the plugin to tweak the *"ReflectionProbe Intensity"* slider within VaM.
- You can rotate the *CustomUnityAsset* atom on the Y axis to rotate. This will automatically sync the *"Skybox Y Rotation"* slider in the your *"Scene Lighting"* settings. Do NOT rotate on X or Z axis!
- You can use the *"Skybox Y Rotation"* slider as well for rotation, then it will only rotate the ReflectionProbe inside the prefab, not the entire *CustomUnityAsset.* However, ReflectionProbe and VaM global illumination will still be in sync.

High Resolution Skies

You can import high resolution skies, for example from hdrihaven.com, with these import settings, see left image below. Make sure to disable *"Generate Mip Maps"* as it would cause a visible seam and increase the *"Max Size"* as needed.

Then you need to create a prefab that contains a <u>large</u> sphere primitive, see right image. You can create a sphere object by using *GameObject* → *3D Object* → *Sphere.* I'm setting the scale to get a sphere with 150 meter diameter. By default, you will have a Sphere Collider, you should remove that. As material we use the *Skybox/Panoramic* shader and set it to use our sky texture.
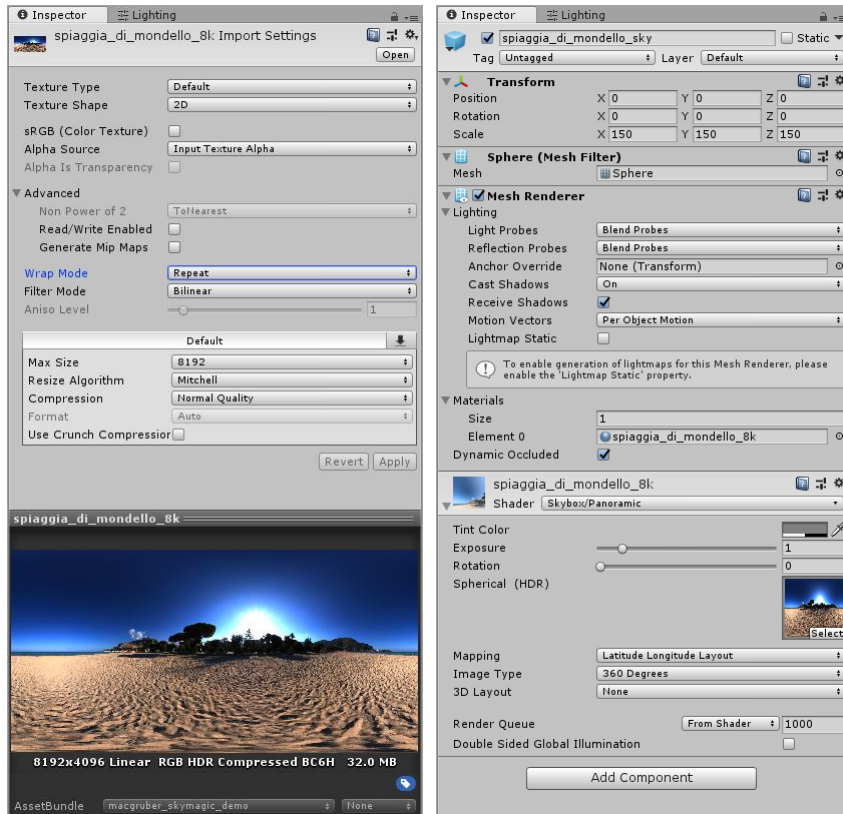
The prefab then needs to be exported as AssetBundle. For the SkyMagic demo I included my sphere prefab inside the same AssetBundle also used for the *ReflectionProbe* prefabs. Note that for some reason the AssetBundle browser does not automatically include the referenced texture file in this case, you might have to check and include it manually, if needed.