CUBEMAPPER **FOR UNITY 4.0 and higher**

Version 1.4, created by Rainer Liessem (Spreadcamp.com)

Index

1. WHAT'S THIS ABOUT?

The Cubemapper is an effective way to make performance friendly reflections of your environment. Now you can make great reflections with just a few clicks from within the Unity Basic or Unity Pro Editor. Cubemaps look great and cost almost nothing in performance since they are made out of several seamless images. Use them to enhance the overall quality of your game.

This system comes with everything you need to get started quickly with Cubemap Reflections:

- Easy to use window
- Performance friendly reflections of your environment
- Supports Unity & Unity Pro (4.0 and higher)
- Culling Mask Support to exclude layers
- Define the resolution (32x32 up to 2048x2048)
- Make customizations by saving PNG Images. Use inbuilt protection features to keep your changes safe.
- Optional real-time switching between Cubemaps using nearest node
- Optional one-click assignment of Cubemaps to your objects, child objects and/or multiple materials on your object

2. THANK YOU

Wow, amazing, you bought our package! We can't even begin to tell you how awesome that is. Like most Indies we're notoriously broke, so we really feel a need to express our thanks here for your purchase. Thanks to you we now can buy more coffee, pizza, or whatever else we commonly live on. But as if plain surviving isn't enough, your purchase also is incredibly motivating to us to do more and to do better, so we hope you will watch out for our future packages as well!

The point is: You rock! We hope that this package will help you and your project(s) every bit as much as you hoped it would when you purchased it. If you feel like something is missing, just contact us and we'll see if we can add it. If you got any problems, please also contact us and we'll do our best to help you out as soon as possible.

It would be great if you could leave a honest review with your rating so that other customers can make a better purchase decision. We're of course always happy to hear positive comments, but we encourage you to be honest. Just don't forget that we really want to help you if there is a problem, so if there is something wrong why not write us an e-mail about what upsets you and maybe we can fix it together?

We're here if you need to contact us, simply write an E-Mail to support@spreadcamp.com and we'll get back to you as soon as possible. Please include a proof of purchase (like a Invoice number / OrderID) along with your request.

Thank you for your support and the best of luck with your projects!

3. INSTALLATION

Simply import the asset package into your project and you are ready to go! But how do you do this? It's easy:

Purchase from the Unity Asset Store:
- Simply click the download button and let it import to your project. Done!

Downloaded .unitypackage:
- Go to the top bar where you can see "File", "Edit", "Assets" etc.
- Click on "Assets" -> "Import Package" -> "Custom Package"
- Find and select the Unity Package file you downloaded to your hard drive. It should look something like *Cubemapperv1.3.unitypackage*
- Press "Open"
- A new Window inside Unity should open up showing you the contents of the Package. Make sure everything is checked and then press **"Import"**
- Done!

4. GETTING STARTED

1. First make sure you have the scene in which you want to make Cubemaps open. Open it now if you haven't done so already.

2. Now please open the Cubemapper System by clicking: **"Window" -> "Cubemapper"**

You should now see the Cubemapper Window together with a help text informing you that we have no Nodes in our scene yet. Let's proceed to the next Step where we add a Node.

PRO-TIPP: You can dock the Cubemapper Window anywhere inside the Editor to fully integrate it with your workflow.

3. The Cubemapper uses Nodes to tell at which position in your 3D-Environment a Cubemap should be generated from. Let's make a Node now by clicking the "New Node" Button in the Cubemapper Window:

Now a new Node will be created at the center of your currently active viewport.

4. If you want, you can select your new Node and reposition it to a more meaningful location. Like a little above a building surrounded by a terrain, or in the middle of your enclosed room. Whatever suits your demands best.

5. Let's generate some Cubemaps! The Cubemapper Window should show you plenty of new Options by now (see Fig. 1). First let's skip briefly over what each Menu is for:

Build: You go here to generate Cubemaps and configure what they will be like.
Assign: Here you can use the One-Click Assign Feature that the Cubemapper offers. It's a Advanced feature that we will talk about later.
New Node: You are already familiar with this ;)
Tools: Some extra tools that could help your workflow when working with Cubemaps. Take a look and decide for yourself if you can use them or not.
Settings: Here you configure Settings specific to the Cubemapper System, such as where new Nodes should be spawned and which path to use for the generated Files.

(Fig. 1)

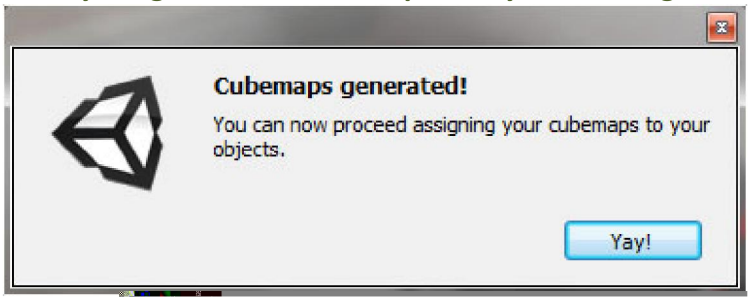
For now let's just make a Cubemap. Make sure you are in the "Build" Menu (as seen in Fig. 1).

Configure Settings:
For now you can leave everything alone and just set the Resolution to *256x256*. Resolutions like 64x64 usually are plenty sufficient, but for the purpose of this tutorial we set it a little higher so you can better see the results.

PRO-TIPP: You can also use the advanced options to do things like generating PNGs for Customization, or use the Culling Mask to exclude Layers from being visible on your Cubemaps.

6. Click the **"Generate Cubemaps!" Button** which will start playback of the game. Don't interrupt this process until it is completed, unless nothing is happening.

7. **If everything worked out as expected you should get this popup:**



Or if you don't see this (for example because the Game ran maximized), then watch out for this in your Console and stop Playback manually:

8. ## Congratulations! You made a Cubemap! :-)

But what about assigning them?

Well, the System comes with a few convenient ways to do that, too. Read about that in the next Section, ***"OPTIONAL STEPS"***

After that, you might also like to check out ***"Tips & Tricks"*** for some additional pointers.

Your Cubemap will look something like this

Where are my generated files?

By default your generated files go to these folders in this format:
Cubemaps: Assets/Cubemapper/Generated Cubemaps/*SceneName - Node Name*.cubemap
***PNGs** (6 per Cubemap):* Assets/Cubemapper/Generated Cubemap Textures/*SceneName - Node Name - Direction*.png

DID YOU KNOW? Since Version 1.3 you can configure your own Output Folders. Simply open the Cubemapper Window and click on "Settings" to change them. Make sure your new Output Folders point to a folder inside your Projects "Assets" Folder. It should in the end look something like this: "Assets/MyCubemapOutputFolder"

OPTIONAL STEPS

The following steps apply only to advanced use cases that make use of features that are optionally available within the system.

Making and using PNG Images for Customization

Please refer to "7. Customizations / PNG Images" further down in the manual.

One-Click Automatic Assignment

Requires: Generated Cubemaps, Nodes (see earlier steps), Cubemap Users (explained in the following steps)

This system allows you to make use of the easy "One Click Assignment" function. In order to use it, you will have to tell the system which objects to consider for assignment. We do this by making these objects what we will call *Cubemap Users.*

1. **IMPORTANT PREREQUISITE:**
   Any object that you want to assign Cubemaps to with this way has to have at least one supported Reflective Shader already attached. Unity ships with Reflective Shaders that are ideal for this (they are what I test this system with, too)

   Make sure the Material on the target object that is to become our Cubemap User is active before proceeding.

   USING CUSTOM SHADERS?
   If you are using a custom shader, then you need to ensure the Shader has a "_Cube" property for the Cubemap Slot, or else it won't work.

2. In the Cubemapper Window go to the Menu "Assign" by clicking this button:
   

3. Select one or more Objects that you want to make a Cubemap User out of.

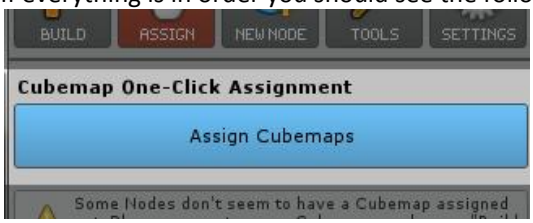
   Note: It's usually enough to select the Parent Object (Root), as the system automatically goes through the children and/or multiple materials to look for supported Shader Materials. Specifically, it looks for

4. Press the button 

in the Cubemapper Window. Your selected objects now are Cubemap Users.

5. Make sure you have generated Cubemaps before proceeding with the next step.

6. If everything is in order you should see the following button. Click it now.



NOTE: *Ignore the warning under the button, it's a false-positive appearing at this moment because we really have not yet assigned anything. The warning is intended to let you know if somehow by accident one of your Cubemaps isn't found anymore.*

7. Clicking the button will cause the system to assign Cubemaps first to your Nodes, and then go through your Cubemap Users to see which Node is nearest to them and assign the Cubemap stored within that nearest Node to its own materials.

8. ### All done! :-) Or is it? Well, there's more Features you could use!

With Cubemap Users you can do even more things, still! Just select the Cubemap User to call up more Options in the Inspector. You might be interested in the following options:

For **Real-time switching** of Cubemaps between the nearest Cubemap Nodes, you'll want to check the "Realtime Switching" Checkbox. But be aware of performance. The effect is also fairly instantaneous, though I hope to improve that in the future somehow (just have to figure out a nice way to do that first) :-)

For many **Objects sharing the same reflective material**, you will want to use the **"Startup Swap"** Checkbox in order to ensure they are using the right Cubemap relative to their position (please read more about that on ***"6. About Shared Materials..."***)

5. TIPS & TRICKS FOR GENERATING GREAT CUBEMAPS

Here's a few things to help you along with making nice Cubemaps:

- The Cubemapper Window remembers your Generation settings, so make sure everything is the way you want it to be before generating Cubemaps. It does not remember any settings for new Nodes and Cubemap Users, though.
- Cubemaps are static, that means they never change, they are essentially like a 360° photograph being displayed on the surface of your objects. This means any dynamic objects like your Player will show on them unless disabled either manually or by using the Culling Mask Settings.
- During Cubemap Generation the Cubemapper automatically disables all other cameras found inside your scene, so if that should give you any errors make sure any scripts that rely on your

camera are disabled before Cubemap Generation.

Did you know that you can use Cubemaps as a Skybox? To do it simply make a new Material and select the Shader "RenderFX/Sky Cubed". If you use a Cubemap as Skybox then I recommend a higher resolution on your Cubemaps, for example 512x512.

A low resolution to your Cubemaps is often better than a high resolution. Higher resolutions have a much bigger file size and their benefits are negligible in most cases. Personally I like a 64x64 Cubemap and I rarely go higher than 256x256. Higher Resolutions would also take up more memory, so why do it if there is little to be gained?

Don't be afraid to experiment with the Reflection Colors on your Material. Sometimes less is more and a really dark reflection color can look better because it highlights bright spots and light sources more than the rest of the environment.

Make sure your Nodes aren't clipping into objects to avoid clipping fragments being visible at your generated Cubemap.

You can place your Nodes wherever you like, but this is what worked well for me using ground-level objects like Bottles or Cars standing on eye level to the Player:

Inside a standard four-walled room, a single Node at the center of the room seems to do the trick. On Outside Scenes, placing the Nodes a little higher up to give the Cubemaps a nice panorama. Feel free to experiment around on your own.

To exclude Objects from showing up on your Cubemaps, you might want to modify the Culling Mask Layers by expanding the "Advanced Settings" in the Cubemapper Window.

6. ABOUT SHARED MATERIALS...

NOTE: *In case you don't use the Cubemap User Script / Features you won't be able to use these convenience functions in the first place, so you can just skip this.*

The Cubemapper has NO problem with "having many Cubemap Materials on one object and/or children", that's supported just fine by the system. Rather this means that you may get confusing results if all of the following conditions apply to you:

[X] *You have more than one Node in your Scene*
[X] *You have 2 or more of the same objects with the same Material placed far apart in your scene*
[X] *You use the Cubemap User Script on the objects sharing the same material*
[X] *The objects with shared materials are near to different Cubemap Nodes*
[X] *You assign Cubemaps by pressing "Build Cubemaps" in the Cubemap Manager or use Real-time Switching at Runtime*

If all of the above is true for you, then you may experience that both your objects will use the same Cubemap from just one Node, instead of using two Cubemaps depending on which one is the nearest to your object.

It's easy to fix: Please select the offending Object and check the Checkbox ***"Startup Swap".***
This ensures upon start of the game the object with "Startup Swap" checked will seek out the nearest Node's Cubemap on its own and assign it. This creates a new Instance of the Material that is

independent of other Objects of the same type. Repeat this for any Object that needs it, but be aware that new Material Instances won't be batched so you shouldn't overdo it.

7. CUSTOMIZATIONS / PNG IMAGES

To make it easier for you to make customizations to your Cubemaps, we have implemented the ability to generate PNG files while also generating Cubemaps. Since Version 1.3 there is also a tool to extract PNGs from already existing Cubemaps. There are a few important things that you should know, so we'd like to ask you to read this full section in order to avoid problems later on.

Making PNGs alongside Cubemaps during Generation

To activate PNG Generation select the Cubemapper Window , bring up the "Build" Menu and expand the "Advanced Options". Now simply check the "Generate PNGs? (slow)" checkbox. Configure the other settings any way you like and just hit the "Generate Cubemaps!" button as usual, except this time it will also create PNGs for you.

By default PNG Files are saved into the following Folder in this type of formatting: "Assets/Cubemapper/Generated Cubemap Textures/*SceneName - Node Name -Direction*.png"

The path may differ if you defined your own output path in the Cubemapper Settings.

Be patient: Making PNGs can take a long time. They have to be re-imported as a usable Asset by Unity, which can be a long process. Especially if you use many nodes in your Scene it will take some time, because each node consists of 6 Images.

Extract PNGs from existing Cubemaps

If you already have a Cubemap and don't want to make PNGs while generating them again you can go to the Cubemapper Window, click on "Tools" and then click on the Button "Extract PNG from existing Cubemap". A new Window should open with a number of steps to follow. You can even define which faces should be extracted, which is a feature unique to this tool.

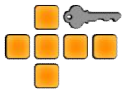
Simply altering and saving the PNG Image isn't enough, so you have to select the Cubemap and drag & drop your modified PNG Images onto the corresponding material slot at your cubemap. The PNG Files are named according to the Slot they should be in. For Example you might have a PNG File like this "MyScene - Cubemap Node 1 - PositiveX" and in the Folder "Generated Cubemaps" a Cubemap named "MyScene - Cubemap Node 1". After you made some changes to the image "MyScene - Cubemap Node 1 - Positive X.png" you have to open the Cubemap in the Inspector and drag the Image File to the Material Slot called "Right (+X)". Similarly you might have made modifications to *"MyScene - Cubemap Node 1 - NegativeY"* which would correspond in your cubemap to the *"Bottom (-Y)"* material slot.

Do you see the pattern here? It goes on like that. The names of both the PNG files and the Cubemaps Material Slots will help you identify which image belongs to which material slot.

WARNING - Change can be overwritten: Generated PNGs and generated Cubemaps will always be overwritten upon the next generation unless you tell the system not to! To avoid overwriting, please find the Node corresponding to your custom Cubemap/PNGs and select it. Then in the Inspector under "Overriding Options" you will want to uncheck "Generate PNGs" and "Generate Cubemaps". Assuming you did it right - and I didn't mess up with the programming ;-) - this Node will now be skipped on any future generation of Cubemaps or PNGs, and therefore your Customizations should be safe.

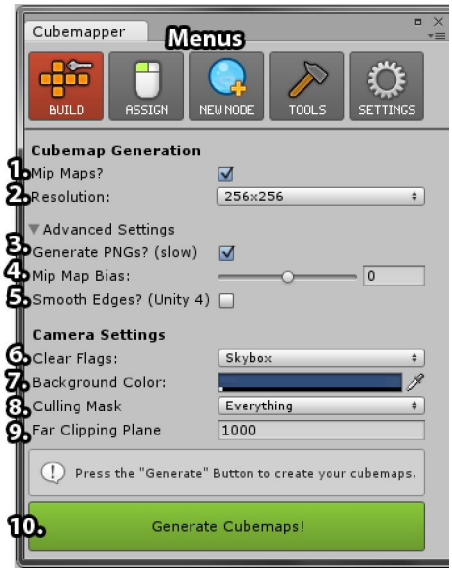
To revert it back and allow generation (and overwriting) again, please find the Node again and check the two checkboxes again. I recommend you write somewhere down which Nodes you altered, so you can find them easier in the future when you need it again.

8.COMPONENTS & VARIABLES (OVERVIEW)



Cubemapper Window

From here you manage the entire system as explained earlier throughout the manual.
NOTE: *The Cubemapper Window saves the last Settings used, so when opening it again make sure everything is the way you want it to be before you press that Generate button!*



The Menus

Build: You go here to generate Cubemaps and configure what they will be like.
Assign: Here you can use the One-Click Assign Feature that the Cubemapper offers. It's a Advanced feature that we will talk about later.
New Node: Adds a new Node to the Scene
Tools: Some extra tools that could help your workflow when working with Cubemaps. Take a look and decide for yourself if you can use them or not.
Settings: Here you configure Settings specific to the Cubemapper System, such as where new Nodes should be spawned and which path to use for the generated Files.

Build Menu in Detail

1. Checking this will enable Mip Maps on the generated Cubemaps

2. Set the resolution of your Cubemap faces anywhere from 32x32 up to 2048x2048 (use with care)

3. Checking this will output 6 PNG Images per Node for Customization Purposes, unless a Node has prohibited PNG Generation (see Nodes in this Overview List). Alternatively you could use the PNG Extract Tool in the Tools Section to extract them from an already existing Cubemap.

4. Set the Mip Map Bias here. This really is meant more for experienced people, but feel free to toy around with it.
DEFAULT VALUE: 0

5. (Unity 4 and higher) Use the Smooth Edges feature if you have problems with the seams in your Cubemap, especially when

using lower Mip Map Levels. It will attempt to smooth out the Edges by a width in Pixels that you can define yourself. **DEFAULT VALUE:** 1

6. Advanced Users can set the Clear Flags if they require, though you will rarely need this and could just keep the setting at "Skybox"

7. Depending on the Clear Flags set before you may want to change the Color. Most Users will be fine with just leaving it the way it is.

8. Defines which Layers should be visible on the generated Cubemaps. Useful to make things like the Player disappear from them.

9. How far of an area can the Cubemaps capture before it clips away? (Default: 1000. Also, Skyboxes are usually always visible so you might not have to change this unless you have a really vast Landscape or something like that)

10. Proceed to generate Cubemaps based on the settings above

Cubemap Node

Cubemap Nodes are used to tell the Cubemap Manager where it should position it's Camera, which is temporarily created during the Cubemap Generation Process to take the 360° screenshots needed for making an Cubemap.

Nodes are used by Cubemap Users as well, with the purpose of holding a reference to the cubemap belonging to this Node that the Cubemap User can reference for use of Real-time switching by proximity to a Node in the Scene.

Generate PNGs *True on default.*	Can exclude this Node from generating PNGs by setting this to false. This is useful only to people that customized generated PNGs for application on Cubemaps and don't want their changes overwritten the next time they generate Cubemaps with PNG Files.
Generate Cubemaps *True on default.*	Set to false to exclude this Node from Cubemap Generation. This is useful only to people that customized Cubemaps and don't want their changes overwritten upon the next generation of cubemaps.
Allow One-Click Assign *True on default.*	When true, this will allow the Cubemap Manager to update the "Used Cubemap" Field when you click the "Build Cubemaps" button. If you customize Cubemaps (especially when you renamed them to something else), you might want to set this to False to prevent errors and/or loss of your changes.
Override Resolution *False on default.*	Set to true to enable overriding the Resolution of Cubemaps generated from this node. It doesn't change cubemaps that already were generated, only the resolution during generation. Most people won't need to use this, but it's here in case you do :-)
Resolution *32x32 on default.*	See "Override Resolution"

Used Cubemap.	This is a field holding a reference to the Cubemap that this Node should be using. This is accessed by Cubemap Users for a fast and performant switching between Cubemaps.

Cubemap User

Cubemap Users are Objects that you made able to use the convenience functions provided within this system, such as one-click assignment of your generated Cubemaps based on nearby nodes, or real-time switching of Nodes during playback.

Small Reminder: An Object that you set to be a Cubemap User requires a Material with Cubemap Support attached, like the Reflection Shaders included with Unity.

Startup Swap *False on default.*	When checked, this Cubemap User will upon Startup seek out the Cubemap assigned to the nearest Node and apply it. This will however create a new Material Instance, so disable this if that is not desired. This is independent of the Realtime Switching, because it is only executed once. This function is intended for cases where you placed several objects sharing the same Material and want them all to use the Cubemap that is nearest to them (see "6. About Shared Materials" for more information)
Realtime Switching *False on default.*	Set to true to enable realtime switching of Cubemaps between the Nodes nearest to this Object. Please make sure you generated and assigned ("Build") Cubemaps first before making Cubemap Users. Realtime Switching is instantaneous. Soft blending between Cubemaps would have only worked with special Shaders and unfortunately I don't know much about shader programming, so this is the best I could do for you.

9. ADDITIONAL RESOURCES

Following are some useful resources to further expand on the usefulness of the Cubemapper System:

Shaders

Transparent Reflective Specular by Amir Abiri *(confirmed to be Public Domain, thanks Amir!)*

10. CONTACT

If you'd like to contact us for whatever reason, this is how you can do it:

Website: http://www.spreadcamp.com
E-Mail: support@spreadcamp.com

If you are a customer and require help, please provide us with a proof of your purchase (such as an Invoice Number / OrderID) so we can make sure you really bought it!

Thank you for your support and the best of luck with your projects!

11. CHANGELOG

v1.4.3
- SUPPORT: Minimum Version now is Unity 4.0
- NEW: Version displayed at "Settings"
- NEW: Ability to set Linear Space
- PRO ONLY: Cubemaps are now generated without going into playmode first. Unity Basic still works, too, but still has to go to playmode to capture the scene.
- FIX: Property Applier Tool now displays if Mip Map selected on cubemap
- MISC: Improved help texts

v1.4
- FIX: Unintentional Cubemap Generation when pressing Play after generating them once
- FIX: Extracted or Generated PNG Images now don't save as compressed anymore
- NEW: Camera Clear Flags and Camera Background Color Settings
- NEW: Smooth Edges Setting (Unity 4 only)
- NEW: Mip Map Bias Setting (use with care)
- NEW: Experimental Tool to apply settings such as Smooth Edges or Mip Map Bias to already existing Cubemaps. Use it carefully and respect the Info Boxes.

v1.3
- SUPPORT: Unity 4 compatible
- FIXED: Possible Memory Leak, especially when using a large number of Nodes
- FIXED: Bug where Settings would not be saved in registry
- NEW: Improved UI with menu buttons
- NEW: Define a custom folder as Output Folders for Cubemaps and PNGs
- NEW: Nodes are now multi-editable
- NEW: Tool to extract PNGs from any existing Cubemap

- NEW: New Nodes now spawn by default in the center of the currently active Viewport. You can configure the Distance to Camera in the Settings. You can also revert back to having Nodes spawn at 0,0,0 if that's what you desire.

v1.2

- Fixed a error in orientation of the generated Cubemaps (many thanks to sjm tech!)
- Now using the faster Color32 where possible
- Info Boxes on the Node Inspector to give a better indication what the settings there are for

v1.1

NOTE to v1.0 Users upgrading to v1.1: *Due to the a big change in the way the system works your old nodes will be lost and have to be re-created. I apologize for the inconvenience.*

- Simplified Interface by putting much of the functionality into a easy-to-use dockable Window
- Fixed an compiler error when using Webplayer Platform
- New Advanced Option: Set Culling Mask (useful to exclude things from the Cubemaps)
- New Advanced Option: Far Clipping Plane
- Using EditorPrefs to keep track of last settings

v1.0

- Initial Release