mayaToCorona
Manual
Version 0.1

Haggi Krey

February 2014

Last Update: 02/2014

www.openmaya.de

Introduction

This is the manual for the maya plugin mayaToCorona. It is created with a open source framework available on github. It implements the great renderer Corona, created by Ondřej Karlík. You can see more information about the renderer on its website.

The plugin is only available for Windows at the moment, and it is tested on Windows7 only. It may work on other version, but I didn‘t check it.

I created a google group for discussion about the plugin, so if you have questions, don‘t hesitate to log in and ask for help. The corona website has a forum as well where you can post questions about the renderer and mayaToCorona, have a look here. I‘ll check both forums as often as I can.

The implementation of a renderer is quite a lot of work, especially the implementation of shading networks. Unfortunatly we have a problem here. In 3dsmax or Cinema4d you can sample the shading network for every rendered sample during rendering. It may be not the very fastest way to use textures and other nodes, but it works fine. Maya does not offer such an approach on a per sample basis. So the texturing is quite limited in maya at the moment. We will have to wait until Corona offers a complete shading network approach like OSL.

But let‘s see how it works. This manual is an extreme work-in-progress work. I‘ll fill it with content as good as I can.

Have fun.

haggi

Installation

If you read this manual, I suppose you already have downloaded the archive with the plugin it is called mayaToCorona.rar and it located here (link).

Of course you have to unpack it. There is no need to unpack it to a special place, just place it wherever you want. The unpacked directory contains a maya module. This module contains everything you need to run the plugin. To avoid problems, please do not modify the name of the directory, it should be named: mayaToCorona.

The archive contains a few directories:

docs

mayaToCorona

mayaToCoronaExamples

The first directory contains this document. The second one contains some example files to check if the renderer works correctly. The third one called mayaToCorona contains the real Maya module. A module conains everything that is necessary to run the plugin.

Next, you have to ensure that maya finds the module. This is quite simple. Maya reads an environment variable called MAYA_MODULE_PATH. If the path to the corona module is found in this variable, the module is read. To manipulate the MAYA_MODULE_PATH you have several possiblities, but I'll present the easiest one here.

Modifying the Maya env file

In the user home directory you should have a maya directory. In Windows it can be found in a path like this: C:\Users\username\Documents\maya\2014-x64. Here you can find a file called Maya.env. Open it and add this line:

MAYA_MODULE_PATH=*path where you unpacked your module*

e.g.

MAYA_MODULE_PATH=*C:\data\myModuleDir\mayaToCorona*

The path should be the directory where the *mayaToCorona.mod* file is located.

Save it, restart maya and the plugin should be visible in the plugin manager. Now try to load the plugin. Depending on the installations on your machine it is possible that some files are missing. One reason is that often the redistributables for Visual Studio are missing. You can find them with the following links, simply download them and install them:

Redistributables for VS 2010

Redistributables for VS 2012

Redistributables for VS 2013

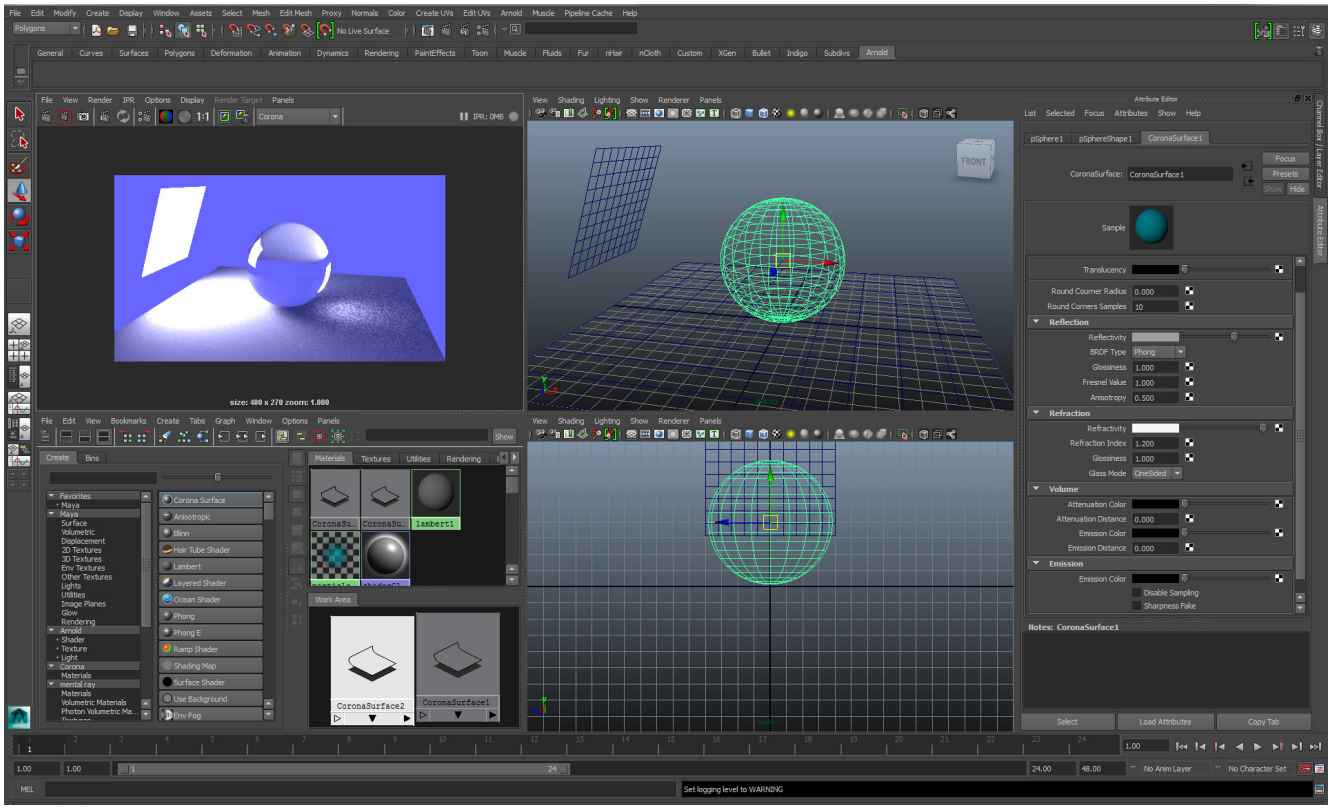
I‘d be happy if you can give me as much feedback as possible about the handling, documentation (as soon as this document is growing to a size to be called a real documentation), and everyting else.

First Steps

Let‘s hope the plugin was successfully loaded. So the next step is to load an example file and render it.

In the mayaToCoronaExamples project, there should be a file called `firstOne.ma`. Open it. If you now open the render window, you should see that the renderer ist set to Corona.

Hit render. The final rendering with scene should look similiar to this one:



If this didn‘t work, something‘s wrong and we will have to find the problem, please find help in the Google Group or the Corona mayaToCorona forum.

Release notes

Version 0.21:

- Added UI diffuse shading feedback for Corona Shader, works now with texture display.
- Added bucket rendering
- Added round corners in shading
- Not allowed textures types are now ignored and an Error message appears. Previously e.g. a hdr as background image caused a crash.
- Moved the init/shutdown lib functions to the pluginMain to avoid unnecessary errors.

Version 0.20:

First public release.

Supported Features

mayaToCorona supports the following features of the corona renderer:

Progressive Rendering

Bucket Rendering

Depth of Field

Transformation Motionblur

Simple texture mapping

Image based lighting

Physical Sun/Sky

Displacement Mapping

Corona surface shader

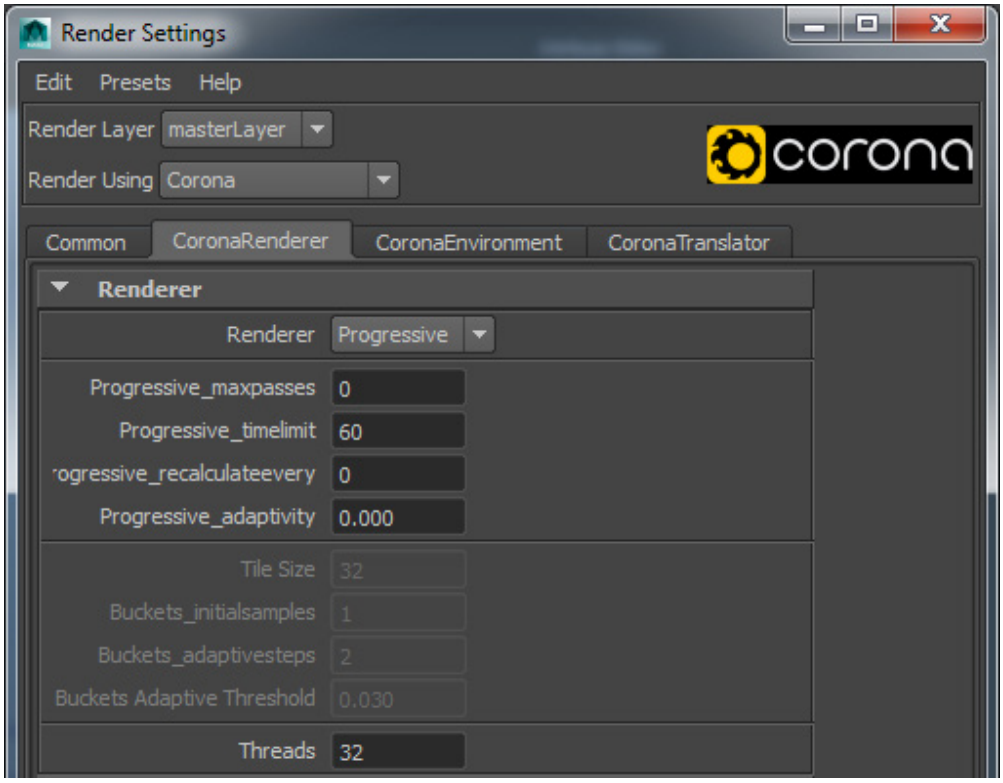
Particle Instancer translation

Supported texture file formats are: jpg, png, bmp and exr.

Render Globals

Renderer

Here you can choose between Bucket and Progressive renderer. Basically every renderer continues to render as long as a certain limit is reached. A limit can be a number of seconds, or a number of samples taken for a pixel.



Progressive

In progressive rendering the limits can be number of passes or a number of seconds passed.

Bucket

Renders the image with buckets.

Features

Displacement

Environment

The environment tab lets you assign a constant color as a background color (default), an image or a physical sun.

Corona Surface Shader

The Corona Surface shader is the default shader. At the moment this is the only available surface shader, but it covers all possible effects for surface shading.

Diffuse Parameters

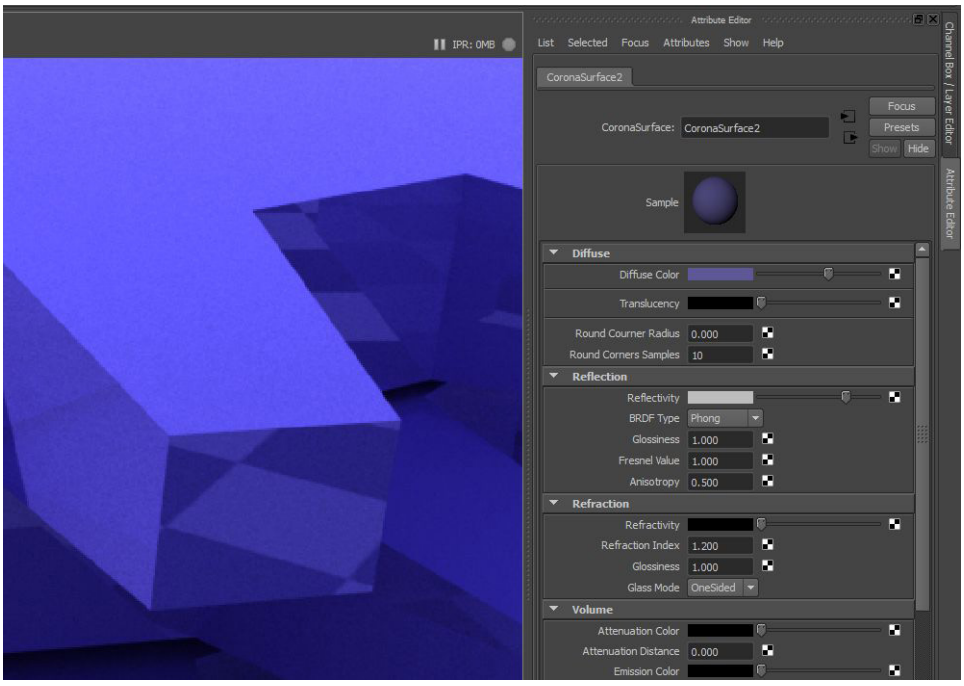
Round Corners

The round corners attribute can create the illusion of round edges for sharp models. This way you do not need to create a small rounding for every small sharp edge, but simply set this values.

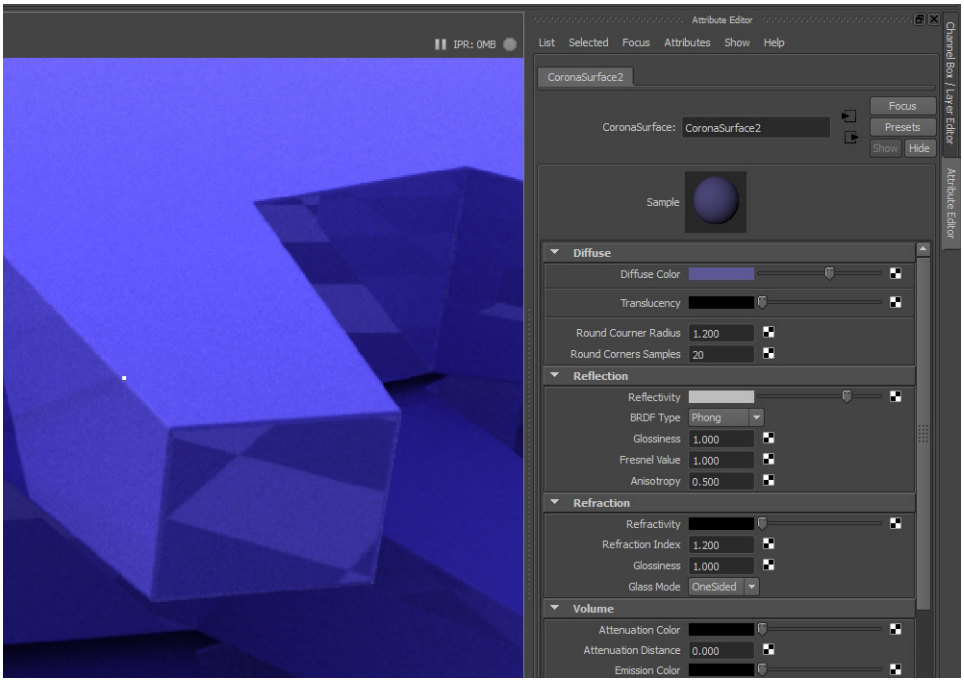
Radius: Sets the Radius for the round corner calculation.

Samples: More samples create a better quality.

Here is an example with no round corners:



And the same rendering with round corners activated:



Textures

At the moment only a few image formats are supported: jpg, png, bmp, exr. All other formats will not be translated.