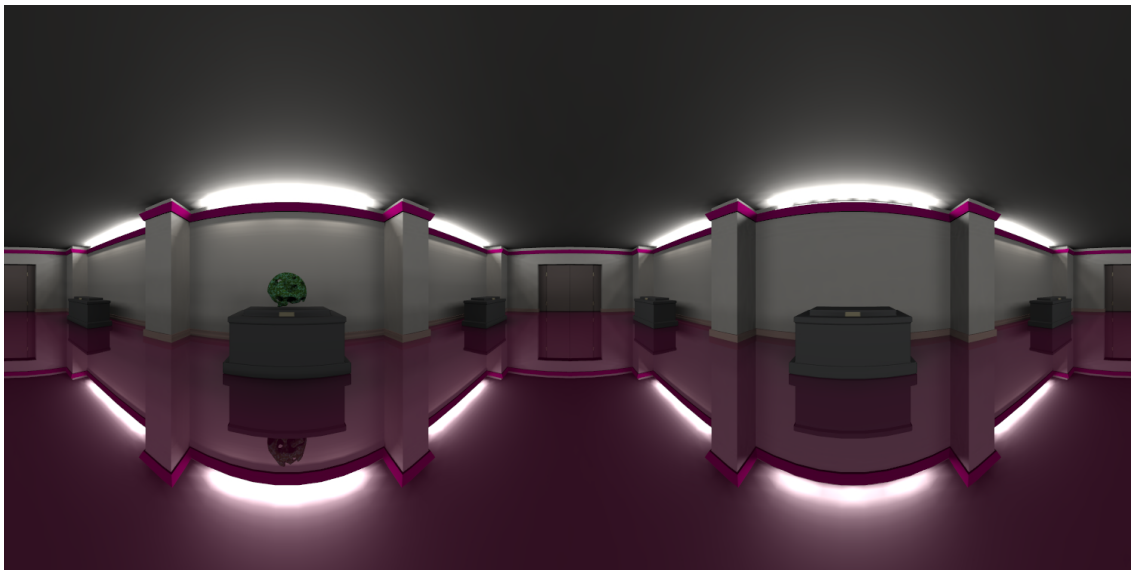


Blender Go Cubic



© May 2005 – Stefano Selleri a.k.a. S68 & macouno

Vers. 0.0.7

‘Guido, io vorrei che tu e Lapo ed io’

Table of Contents

INTRODUCTION	3
PACKAGE CONTENT	3
DEPENDENCIES	3
INSTALLATION	3
USAGE	3
GENERAL SETTINGS	4
THE CYLINDRICAL PANORAMA	4
THE CUBIC PANORAMA	4
OBJECT TURNTABLE	5
LATER OPERATIONS	5
ADVANCED USAGE	6

Discalimer

This software is provided 'as it is' no guarantees expressed or implied are given, except that it will occupy some space on your disk(s). The software, bag.blend and this document are © 2005 Stefano <S68> Selleri and macouno and released under Blender Artistic license which you should have received with the package. If this is not the case a copy of said license is available at www.blender.org or directly from the author selleri@det.unifi.it.

Introduction

Blender Go Cubic (BGC) is a Python script designed to prepare the 360° panoramas, or the 6 square view from a given point necessary to create QuickTime panoramas or the numerous shots for an Object QuickTime Virtual Reality. Actually these latter can be used in 3DNP by thoro. **BGC** itself was jointly created by macouno and myself.

Package Content

The **BGC** package should contain:

1. **BGC.pdf** – This file you are reading ☺
2. **BGC.blend** – An example Blender file with the script within
3. **BGG-0.0.7.py** – The script, as a separate text file.
4. **Licence.txt** – The Blender Artistic License under which this package is given.

Dependencies

BGC needs the following softwares to be correctly installed on your computer to work:

1. An operating system of your choice ☺.
2. Blender 2.36 and maybe later version (© The Blender Fund - www.blender.org).
3. Python 2.3.4 or whatever Python is recommended with your version of Blender (© - www.python.org)

Installation

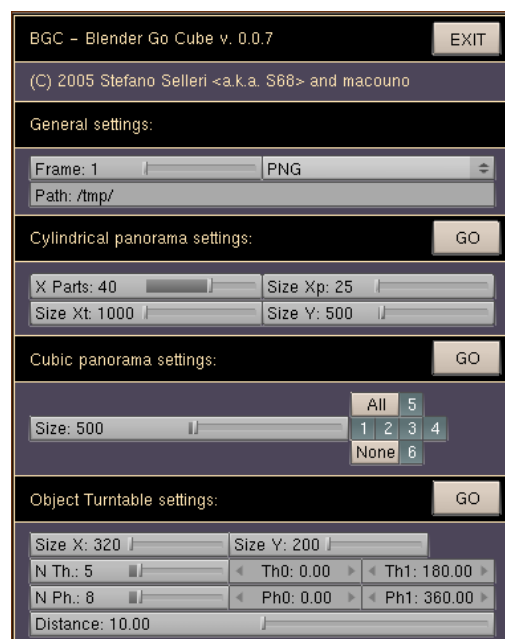
Unpack the package in a directory of your choice.

Once you are done you can load the **BGC.blend** file of this package in Blender, or load the script in a text window, or copy the script file in your Blender installation tree and run it from the Script window.

Usage

Place your mouse cursor on the left Blender window, the text window containing the **BGC** script and press **ALT-P**.

The Graphical User Interface (**GUI**) here on the right should appear. If not you have a problem.

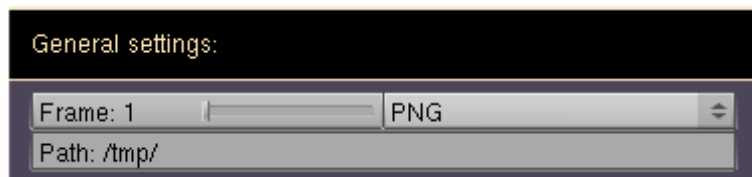


The **GUI** is divided into three part:

1. A top part with credits and one button: **Exit** – The button does exactly what you are thinking of.
2. A central, big, part divided in four subparts, top to bottom:
 - 2.1. a **General Settings** part
 - 2.2. a **Cylindrical Panorama** part
 - 2.3. a **Cubic Panorama** part
 - 2.4. a **Object Mode** settings

General Settings

You must set here the **FRAME** you want to render, the type of image to render (**JPG**, **PNG**, **TARGA**, **PNG**) and the path to store them.



If the path does not end with '/' then the last letter are used as a prefix to the created file name.

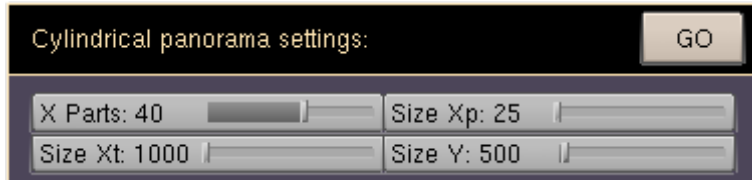
Beware that windoz might prefear 'C:\tmp\' to '/tmp/' or whatever.

These settings will override those in Rendering context. After **BGC** is done the settings will be restored to their original values (hopefully).

The Cylindrical Panorama

BGC with this option creates a single 360° image.

What really matters is the **TOTAL** size along the azimuth (**X**) direction, set via the **Size Xt** slider. This slider, together with the **Size Y** slider defines the total size of the panorama. A 2:1 ratio as the defaults suggests is a good idea.



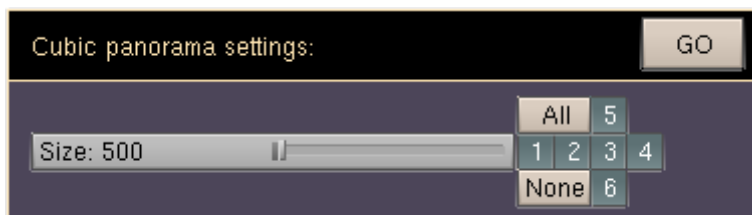
The other two slides defines how many separate images are rendered (**X Parts**) and how large (**Size Xp**) each is. These two are not independent with **Size Xt** and a change in one reflects on the others.

The higher **X Parts** is the better.

The azimuth field of view is always 360°, The vertical field of view depends on **Size Y**, the highest, the greatest the field of view, but if you really want a large view, vertically, you must use the Cubic panorama!

The Cubic Panorama

This option creates six square shot. The only setting is **Size**.



The shots are taken from Camera position and points forward, right, back, left, top and bottom in this order. Camera lenses are fixed to 16.5.

The six shots will be '0001' to '0006' appended to whatever your path is.

Since you might want to render the six shots separately the 6 toggle buttons nearby allows you to specify which view to render on a one-by-one basis.

Object Turntable

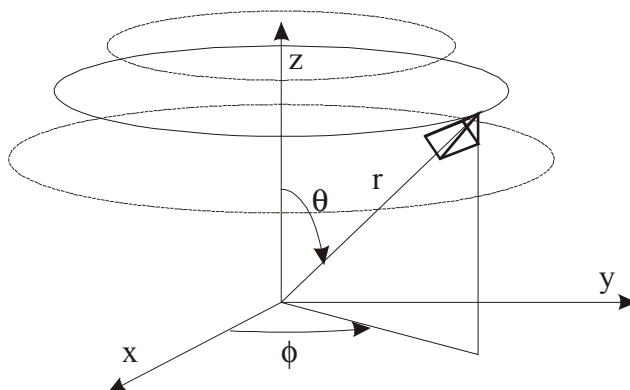
This is the opposite of a cubic panorama. The camera is placed at a given distance from the origin, where the Object should be, and rotates around it taking shots.

SizeX and **SizeY** define the image size. Camera is then placed at points of a spherical reference defined by the other sliders.

Camera moves on **N Th** rings, on each ring it takes **N Ph** shots. Rings goes from **Th0** to **Th1** inclusive. Shots goes from **Ph0** included to **Ph1** excluded.

The distance of the camera from the Origin is fixed and equal to **Distance**.

For example the default settings takes 5 rings of 8 shots each (40 rendered images) at $\theta = 0, 45^\circ, 90^\circ, 135^\circ, 180^\circ$ and $\phi = 0, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$.



Images are saved as Row##shot##, the first number is the ring, the second the shot.

It is worth noting that true good results are obtained with **N Th** = 19 and **N Ph** = 36, that is a 10° step.

Many object can be seen only in the **Th0** = 0 **Th1** = 90° range, usually changing **Ph0** and **Ph1** is a bad idea.

Later Operations

Get GoCubic.exe from its site and use it to convert the single cylindrical or six cubic images to a QuickTime VR panorama.

For an Object turntable you are out of luck, no freeware, but the JavaScript script by thoro <http://www.thoro.de/portfolio/verschiedenes/3DNP.html> handles the images very well too!

Advanced Usage

For Cubic panorama 'forward' is the positive direction of y axis (MUN1 blender view)

If the Camera is parented to an empty then you can rotate the empty and the forward direction stays the one pointed by the Empty y axis.

Parenting is tricky, so test what you are doing! And clear rotations/positions of the camera ;)

The same holds for Object Turntable, The spherical reference is defined by the Empty both in origin and theta-phi angles.