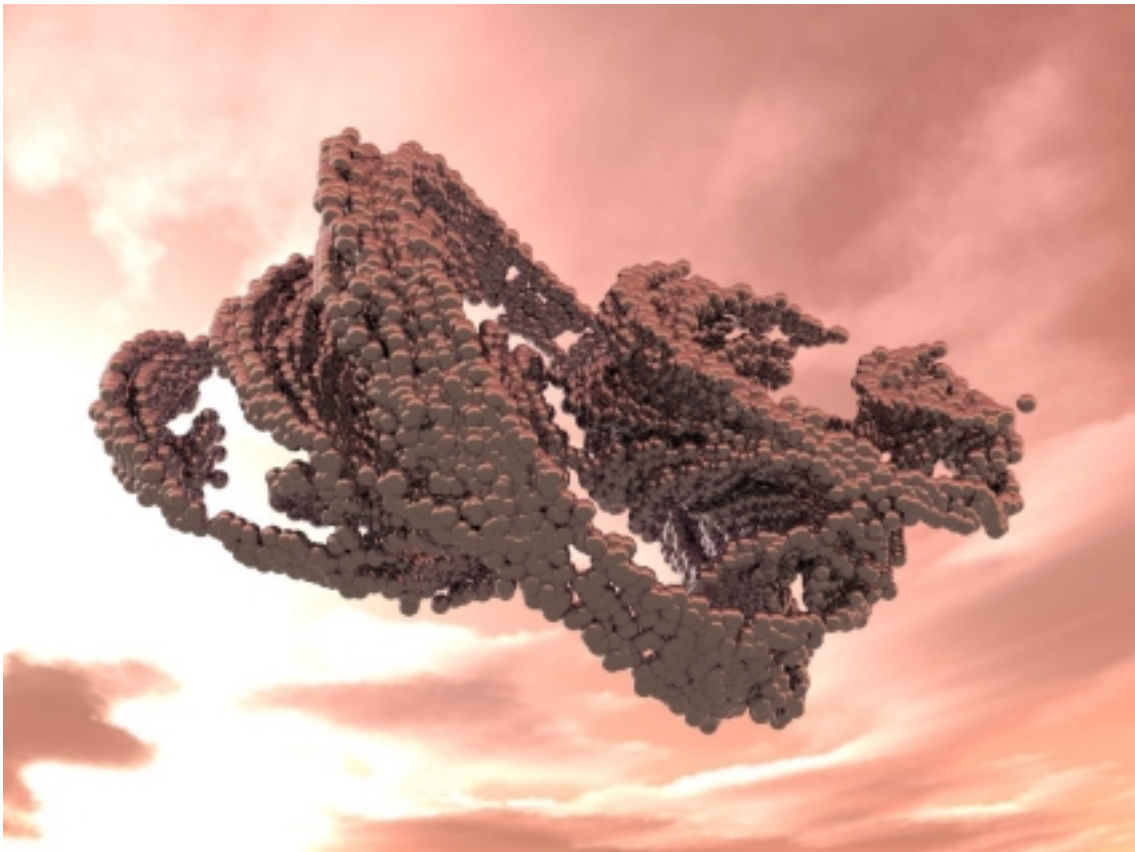


Blender Julia Fractals



© March 31 – Stefano Selleri a.k.a. S68

Vers. 0.0.1

‘Signori si nasce (e io lo nacqui, modestamente)’

Table of Contents

INTRODUCTION	4
PACKAGE CONTENT	4
DEPENDENCIES	4
INSTALLATION	4
USAGE	4
GENERATION SCHEME	5
MAIN SETTINGS	5
LEFT COLUMN	5
RIGHT COLUMN	5
EXECUTION	6
CHANGELOG	6
0.0.1	6
KNOWN BUGS	6
TODO	6

Disclaimer

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, example.blend and this document are © 2004 Stefano <S68> Selleri 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 Julia Fractals (BJF) is a Python script designed to build worlds. Its main purpose is the generation of Julia fractals.

Package Content

The **BJF** package should contain:

1. **BJF.pdf** – This file you are reading ☺
2. **BJF.blend** – An example Blender file with the script within
3. **BJF-0.0.1.py** – The script, as a separate text file.
4. **Licence.txt** – The Blender Artistic License under which this package is given.

Dependencies

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

1. An operating system of your choice ☺.
2. Blender 2.32 and maybe later version (© The Blender Fund - www.blender.org).

Installation

Unpack the package in a directory of your choice.

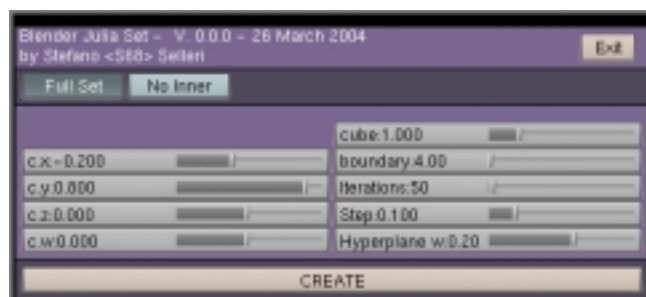
Once you are done you can load the **bjf.blend** file of this package in Blender.

Usage

Place your mouse cursor on the left Blender window, the text window containing the **BJF** scriupt 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 four part:

- 1 - A top part with credits and an **Exit** button – The **Exit** button does exactly what you are thinking of.
- 2 - A Upper middle part with two (for now) radio buttons. These selects the two possible way of generation: a Full Julia Set and a Julia Set where inner points are not generated.
- 3 - A middle part, the bigger, main, part, holding the main settings.



4 - A lower part with just one button: **CREATE** to launch execution.

Generation Scheme

A Julia Set is a portion of quaternions space (4D space) for which the following succession

$$q_{n+1} = q_n^2 + c$$

does not diverge. A quaternion is a member of a 4D space of the kind

$$q = x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}$$

with $x, y, z, w \in \mathbb{R}$ and

$$\begin{aligned} \mathbf{ii} &= -1 & \mathbf{ij} &= k & \mathbf{ik} &= -j \\ \mathbf{ji} &= -k & \mathbf{jj} &= -1 & \mathbf{jk} &= i \\ \mathbf{ki} &= j & \mathbf{kj} &= -i & \mathbf{kk} &= -1 \end{aligned}$$

Main Settings

Left column

The left column allows to set the value of c in the iterative formula. The four real numbers cx , cy , cz and cw are asked for.

Right column

The Right column exhibits more varied settings:

- Cube:** is the side of the cube in which the Julia set is sought for. A value of 1.00 (the default can sometimes chop off ends. Try 1.2 in these cases
- Boundary:** is the limit beyond which the series is considered divergent. If $|q_n|$ is greater than this value the iteration stops and the point is not part of Julia Set.
- Iterations:** if the point $|q_n|$ has not escaped the **Boundary** when n equals this value then the point is considered as part of the Julia set.
- Step:** the space step for points. The cube in which the set is sought for is subdivided into tiny cube this big. Each vertex of the cube is a point, each point is searched for. The default value 0.1 is quite big. A value of 0.01 gives much better results but the created Object easily have more than 100.000 vertices. You are warned!
- Hyperplane:** the Julia set is a 4D object that can be projected onto 3D space. This is done by getting a 3D section of it on an hyperplane characterized by a $w = \text{constant}$ equation.

Execution

When the execution button is pressed the **cube** of search is divided into cubes with **step** sized edges. Then each point is checked for. If it is in the Julia set a vertex is created. This leads to an Object containing a cloud of vertices and is the default behavior.

On the other hand you can choose the **No Inner** radio button. In this case (slower!) each point is checked (Highly unefficiently) if it is surrounded by points which are in the Julia set. If this is true then no vertex is created, effectively generating vertices only on boundaries (and hence less vertices than in the other case!).

ChangeLog

No changelogs before 0.0.6... so new stuff is from 0.0.7 onwards!

0.0.1

New in 0.0.1 – 31-03-2004

- No Inner option

Known Bugs

- None (?)

ToDo

Surfaces!