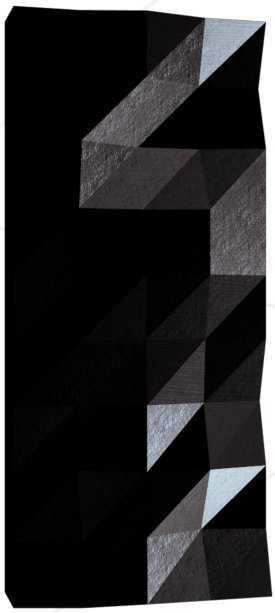


Computer graphics project

fract'ol

42 staff staff@42.fr

Summary: This project is meant to create graphically beautiful fractals.

Contents

Chapter I

Foreword

Here's what Wikiepedia has to say on hydraulic fracturing:

The "hydraulic fracturing," is the targeted disruption of geological formations with low permeability by means of injection under high pressure of a fluid to micro-cracking and crack the rock. This fracturing can be performed near the surface or at depth (over 1 km or more than 4 km in the case of shale gas) and from vertical wells, sloped or horizontal.

This relatively old technique (1947), developed for conventional oil deposits, is renewed by its association with horizontal drilling (developed from 1980). It is the gradual mastery of the economic viability of this association for non-conventional deposits, who guided the recent development of the operation of these: it made available formerly inaccessible resources, or which have been exploited at exorbitant costs and slowly.

It is carried out by fracturing the rock with a "stress" mécanique3 using a fluid injected under high pressure from a surface drilling, to increase the macro-porosity and the micro-porosity. The fluid could be the water, a slurry or a technical fluid whose viscosity was adjusted.

This project is not called *f*ract'oil and accordingly has no relation to hydraulic fracturing.

Chapter II

Subject

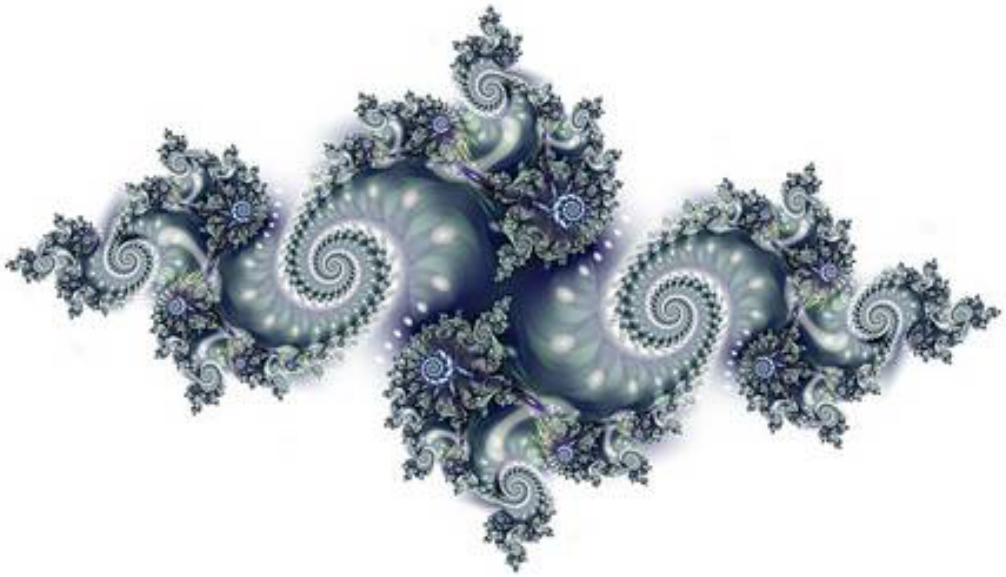
This project is to create a small fractal exploration program. Start by seeing what a fractal is.

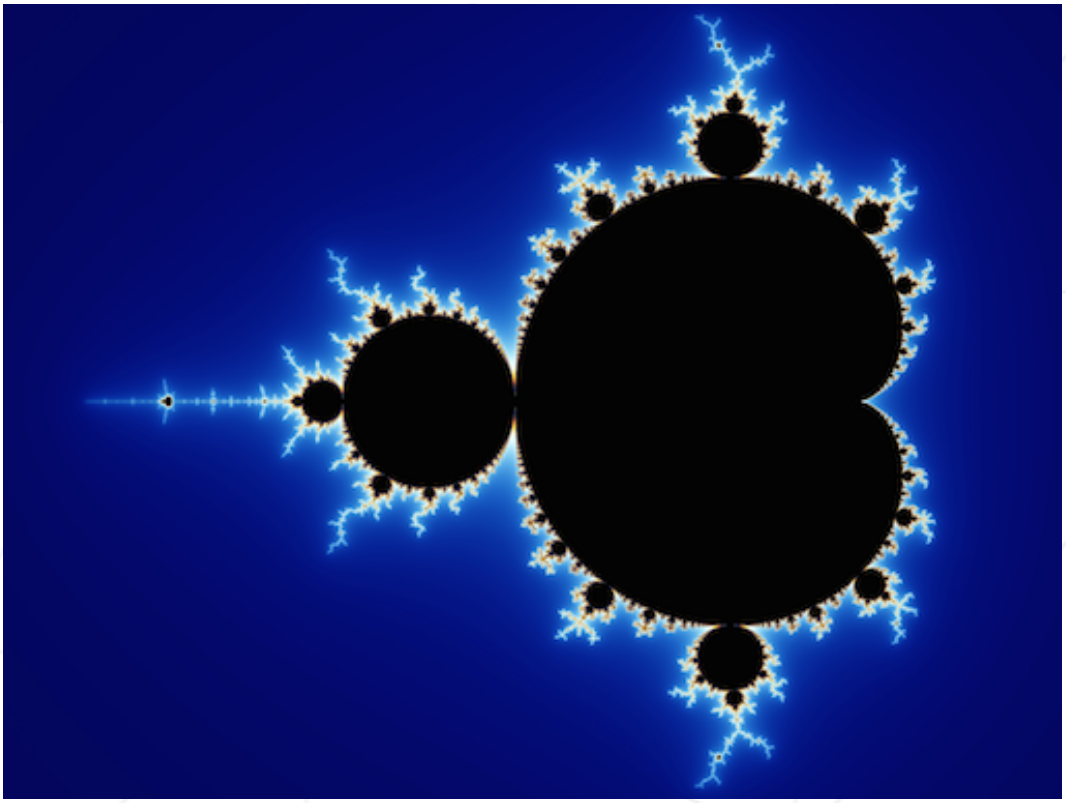
The constraints are as follows:

- Your software should offer at least 3 different types of fractals, from which at least the Julia set and the Mandelbrot set (a third fractal is your choice).
- It must be possible to vary the mouse (without clicking) the parameter of the Julia set. The other types of fractal, it is left to your option.
- The mouse wheel zooms in and out on the current mouse position, and this in a quasi infinite way (modulo the limits of the machine). This is the same principle of fractals.
- There must be a minimum play of color to feel the depth of each fractal. Better hack away on psychedelic effects.
- A parameter is passed on the command line to define what type of fractal is to view. If there is no parameter provided, or if the parameter is invalid, the program displays a list of available parameters and exits.

As regards the graphic representation:

- You must use the `minilibX`, in its version for XQuartz installed by default on the dump, or in its native version for MacOS X.
- You have to manage properly the `expose`.
- ESC will exit the program.
- The use of `images` of the `minilibX` is strongly recommended.





Chapter III

Subject - Bonus

Here are some interesting ideas to achieve bonus. Of course, you can add bonus of your invention, which will be assessed by your checkers.

- In addition to the zoom: displacement with the arrows.
- Make rotate the colour palette.
- Full of fun fractals (sites listing more than one hundred different type).
- 3D representation.
- For two valid parameters in command line, results two windows of two fractals.

Chapter IV

Instructions

- This project will not be corrected by humans. So, fell free to organize and name your files as you wish, nevertheless within the constraints listed here.
- The executable must be named `fractol`.
- You must provide a Makefile.
- Makefile should compile the project, and must contain the usual rules. It must recompile the program only when necessary.
- If you are smart and you are using your own library `libft` for `fractol`, you have to copy the sources files and the associated `Makefile` into a folder called `libft` which should be at the root of your delivery repository. Your `Makefile` shall compile the library, by calling the Makefile and compile your project.
- Same for MinilibX native on MacOS.
- You must not have global variables.
- Your project must be according Norme coding standard.
- You must handle errors in a reasonable way. In any case your program shouldn't exit unexpectedly (Segmentation fault, etc...).
- You have to deliver into the root of your delivery repository an `auteur` file containing your login followed by a '\n':

```
$>cat -e auteur
xlogin$
$>
```

- You have the right to use all the functions you want of minilibX, of your libft, of math lib, and of libC if there is not a function that should be in your libft.
- You can ask questions on the forum, on jabber, IRC, ...

- Success to all of you!