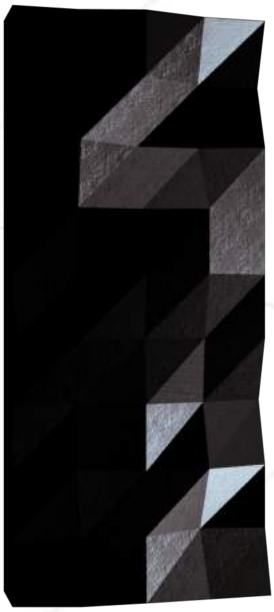


Computer graphics project

fract'ol

Pedago Team pedago@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

Introduction

The term fractal was first used by mathematician Benoit Mandelbrot in 1974,
he based it on the Latin word `fractus`, meaning "broken" or "fractured".
A fractal is an abstract mathematical object, like a curve or a surface, which has a similar pattern whatever the scale.

Various natural phenomena – like the romanesco cabbage – have some fractal features.

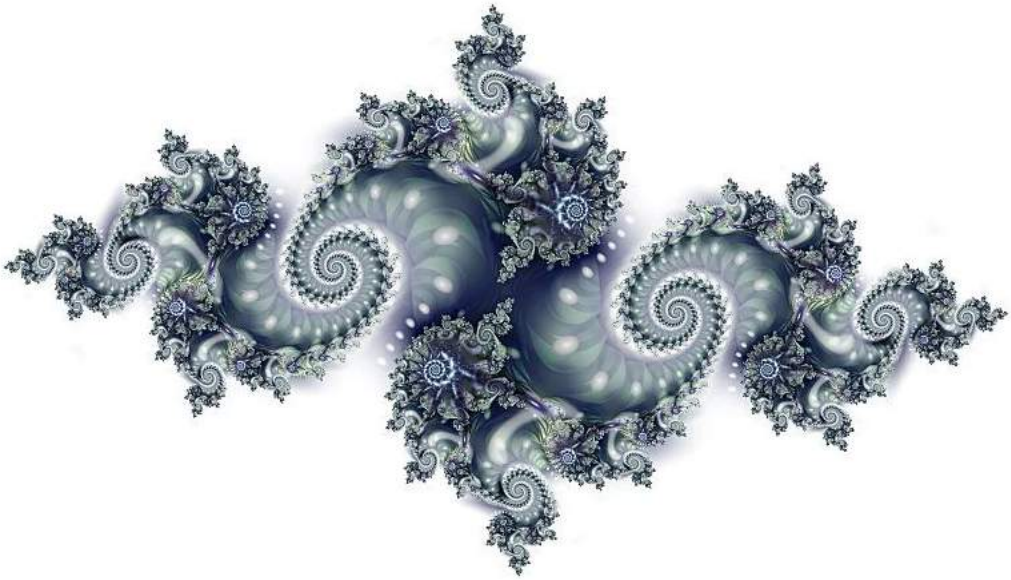
Now, it's your turn to modelize some magnificent fractals!

Chapter III

Objectives

Now that you took over your first graphical library: the `miniLibX`,
it's time for you to light all the pixels of your screen at the same time!

This new project will be the opporunity for you to hone your `miniLibX` skills, to make you discover/use the mathematical notion of `complex numbers` and to take a peek at the concept of `optimization` in computer graphics.



Don't forget to watch the videos on the e-learning!

Chapter IV

General Instructions

- This project will be corrected by humans only. So, feel free to organize and name your files as you wish, but within the constraints listed here.
- The executable file must be named `fractol`.
- You must submit a `Makefile`.
- Your `Makefile` must compile the project and must contain the usual rules. It must not recompile and re-link the program unless necessary.
- If you are clever, you will use your library for your `fractol`. Submit also your `libft` folder, including its own `Makefile` at the root of your repository. Your `fractol` `Makefile` will have to compile the library, and then compile your project.
- You cannot use global variables.
- Your project must follow `the Norm`.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- Your program cannot have memory leaks.
- You'll have to submit a file named `author` containing your username followed by a '\n' at the root of your repository.

```
$>cat -e author
xlogin$
```

- You `must` use the `miniLibX`. Either in the version that is available on the system, or from its sources. If you choose to work with the sources, you will need to apply the same rules for your `libft` as those written above.

- For your mandatory part, you are allowed to use the following functions:
  - `open`, `read`, `write`, `close`
  - `malloc`, `free`
  - `perror`, `strerror`
  - `exit`
  - All the functions defined in the `math` library (-lm and man 3 math)
  - All the functions defined in the `miniLibX` library.
- You are allowed to use other functions to complete the bonus part as long as their use is justified during your evaluation. Be smart!
- You can ask your questions on the forum, Slack, etc.

Chapter V

Mandatory part

This project's goal 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, including the Julia set and the Mandelbrot set. The third fractal (and the optional additional ones) can be whatever you want.
- It must be possible to make the parameter of the Julia set vary only with the mouse (without clicking). For the other types of fractal, the decision is yours.
- The mouse wheel zooms in and out, almost infinitely (within the limits of the computer). This is the very principle of fractals.
- You must use at least a few colors to show the depth of each fractal. It's even better if you hack away on psychedelic effects.
- A parameter is passed on the command line to define what type of fractal will be viewed. If no parameter is provided, or if the parameter is invalid, the program displays a list of available parameters and exits properly.

As for the graphic representation:

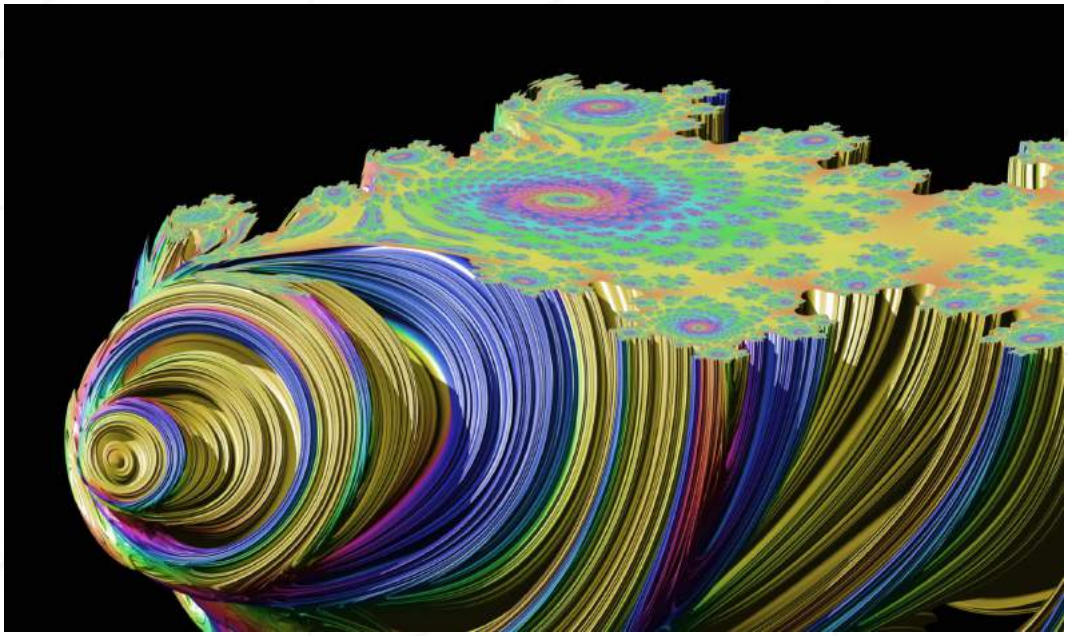
- ESC will exit the program.
- The use of `images` of the `minilibX` is strongly recommended.

Chapter VI

Bonus part

Here are some interesting ideas for the bonuses. Of course, you can add bonuses of your own invention, which will be assessed by your evaluators.

- The zoom follow the actual mouse position.
- In addition to the zoom: moving with the arrows.
- Make the color range shift.
- Multi-threading implementation.
- Plenty of fun fractals (there are more than a hundred different types of fractals referenced online).
- 3D representation.
- Two valid parameters in the command line, resulting in two fractals in two windows.



Chapter VII

Submission and peer-evaluation

Submit your work on your `GiT` repository as usual. Only the work on your repository will be evaluated.