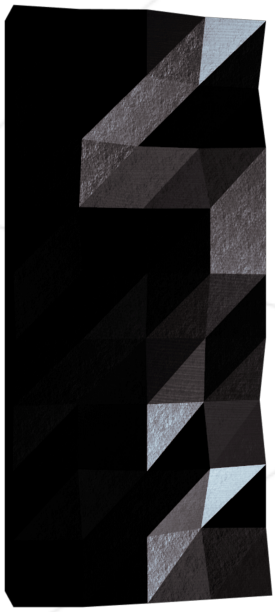


fract'ol

Computer Graphics Fractals

Summary: This project involves creating graphically beautiful fractals.

Version: 3.2

Contents

I Foreword **2**

II Introduction **3**

III Objectives **4**

IV Common Instructions **5**

V Mandatory part **7**

V.1 Rendering 8

V.2 Graphic management 8

VI Bonus part **9**

VII Submission and peer-evaluation **10**

Chapter I

Foreword

Here's what Wikipedia has to say on **hydraulic fracturing**:

"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, which 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 performed by fracturing the rock by a mechanical "stress" using a fluid injected under high pressure from a surface drilling, to increase the macro porosity and less the micro porosity. The fluid could be the water, a slurry or a technical fluid whose viscosity was adjusted.

This project is not called *fract'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* which means "broken" or "fractured".

A fractal is an abstract mathematical object, such as a curve or a surface, whose pattern remains consistent at every scale.

Various natural phenomena, such as the Romanesco cabbage, exhibit fractal features.

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

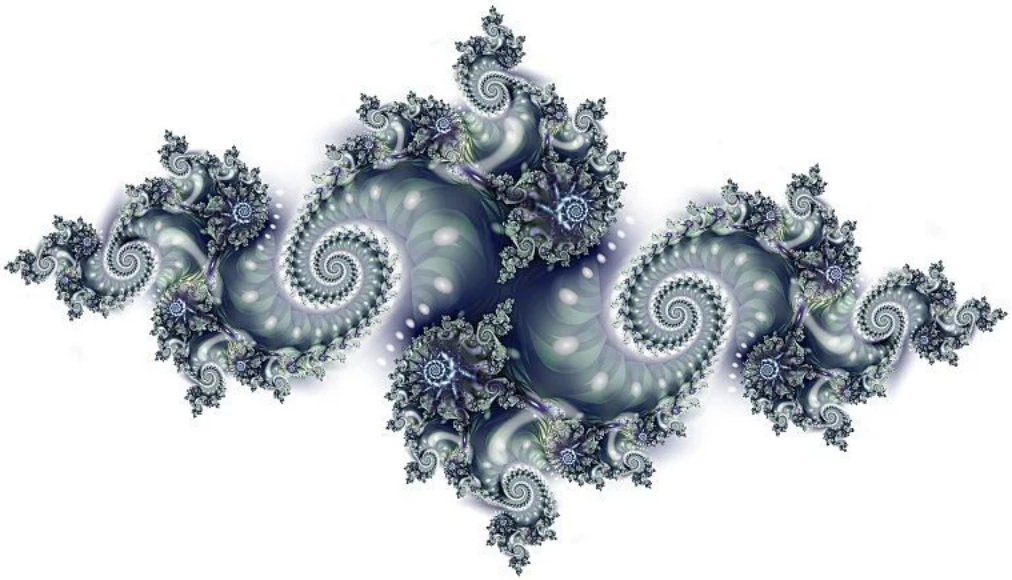
Chapter III

Objectives

It's time for you to create a basic computer graphics project!

You will use the school's graphical library, the `MiniLibX`. This library was developed internally and includes basic necessary tools to open a window, create images and deal with keyboard and mouse events.

This will be an opportunity for you to become familiar with the `MiniLibX` library, discover or use the mathematical concept of **complex numbers**, explore computer graphics **optimization**, and practice **event handling**.



Chapter IV

Common Instructions

- Your project must be written in C.
- Your project must be written in accordance with the Norm. If you have bonus files/functions, they are included in the norm check, and you will receive a `0` if there is a norm error.
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc.) except for undefined behavior. If this occurs, your project will be considered non-functional and will receive a `0` during the evaluation.
- All heap-allocated memory must be properly freed when necessary. Memory leaks will not be tolerated.
- If the subject requires it, you must submit a `Makefile` that compiles your source files to the required output with the flags `-Wall`, `-Wextra`, and `-Werror`, using `cc`. Additionally, your `Makefile` must not perform unnecessary relinking.
- Your `Makefile` must contain at least the rules `$(NAME)`, `all`, `clean`, `fclean` and `re`.
- To submit bonuses for your project, you must include a `bonus` rule in your `Makefile`, which will add all the various headers, libraries, or functions that are not allowed in the main part of the project. Bonuses must be placed in `_bonus.{c/h}` files, unless the subject specifies otherwise. The evaluation of mandatory and bonus parts is conducted separately.
- If your project allows you to use your `libft`, you must copy its sources and its associated `Makefile` into a `libft` folder. Your project's `Makefile` must compile the library by using its `Makefile`, then compile the project.
- We encourage you to create test programs for your project, even though this work **does not need to be submitted and will not be graded**. It will give you an opportunity to easily test your work and your peers' work. You will find these tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to the assigned Git repository. Only the work in the Git repository will be graded. If Deepthought is assigned to grade your work, it will occur

after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.

Chapter V

Mandatory part

Program name	`fractol`
Turn in files	`Makefile, *.h, *.c`
Makefile	`NAME, all, clean, fclean, re`
Arguments	`The type of fractal to display and any other option available`
External functs.	• `open, close, read, write, malloc, free, perror, strerror, exit.` • `All functions of the math library (-lm compiler option, man 3 math).` • `All functions of the MiniLibX library.` • `ft_printf or any equivalent YOU coded.`
Libft authorized	`Yes`
Description	`This project is about creating a small fractal exploration program. First, you need to understand what a fractal is.`

Your project must comply with the following rules:

- You **must** use the `MiniLibX` library. Either the version available on the school machines, or installing it using its sources.
- You have to turn in a `Makefile` which will compile your source files. It must not relink.
- Global variables are forbidden.

V.1 Rendering

- Your program must offer the **Julia** set and the **Mandelbrot** set.
- The mouse wheel allows zooming in and out almost infinitely (within the computer's limits). This is the very principle of fractals.
- You must be able to create different Julia sets by passing different parameters to the program.
- A parameter is passed on the command line to define what type of fractal will be displayed in a window.
    - You can handle more parameters to use them as rendering options.
    - If no parameter is provided, or if the parameter is invalid, the program displays a list of available parameters and exits properly.
- You must use at least a few **colors** to reveal the depth of each fractal. Experimenting with psychedelic effects is encouraged.

V.2 Graphic management

- Your program has to display the image in a window.
- Window management must remain smooth (e.g., switching to another window, minimizing, etc.).
- Pressing `ESC` must close the window and quit the program in a clean way.
- Clicking on the cross on the window's frame must close the window and quit the program in a clean way.
- The use of the `images` of the `MiniLibX` library is mandatory.

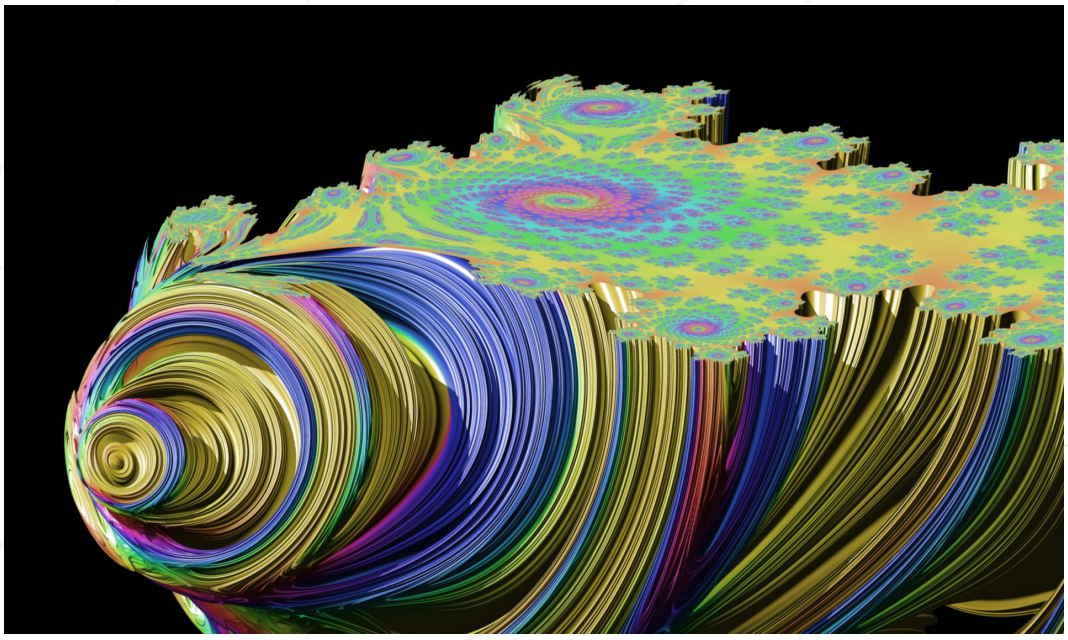
Chapter VI

Bonus part

Typically, you would be encouraged to develop your own original additional features; however, more interesting graphic projects await you in the future. Don't spend too much time on this assignment!

You will get some extra points with the following features:

- One more different fractal (more than a hundred different types of fractals are referenced online).
- The zoom follows the actual mouse position.
- In addition to zooming, allow moving the view using the arrow keys.
- Make the color range shift.



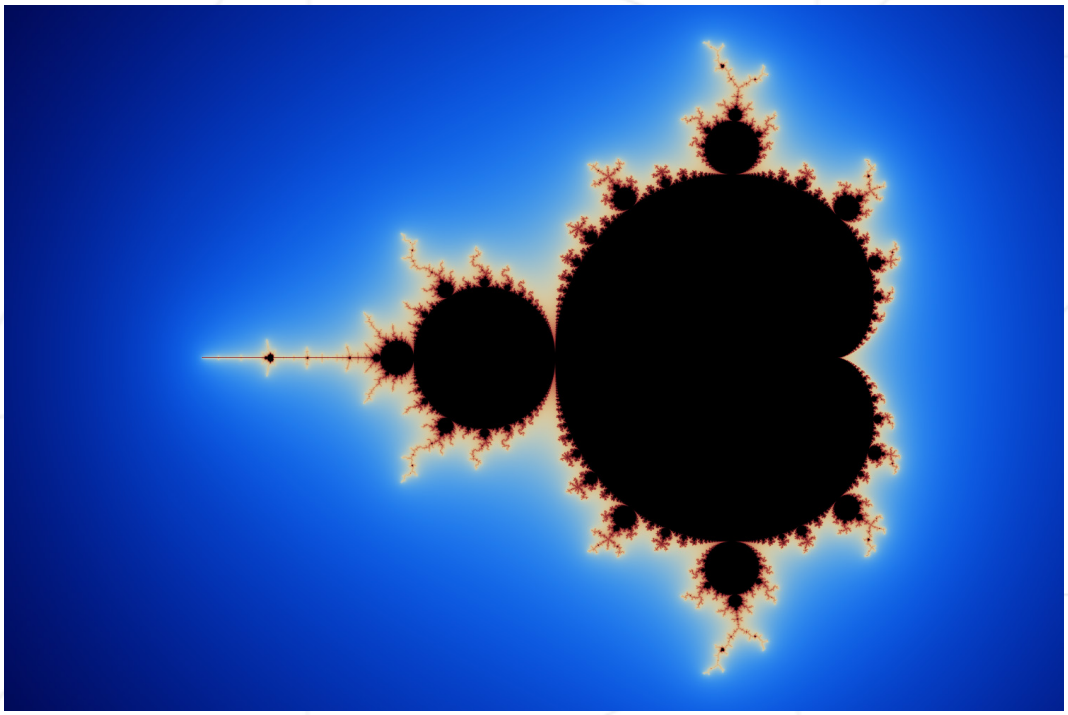
The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter VII

Submission and peer-evaluation

Submit your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your files to ensure they are correct.

As these assignments are not verified by a program, feel free to organize your files as you wish, provided you submit the mandatory files and comply with the requirements.



file.bfe:VAD2sO2qgbqPEXROeASsmsgnY0o0sDMJev7zFHhw
QS8mvM8V5xQQpLc6cDCFXDWTiFzZ2H9skYkiJ/DpQtnM/uZ0