



# Projet d'Infographie

## fract'ol

Pedago Team [pedago@42.fr](mailto:pedago@42.fr)

*Résumé: Ce projet consiste à créer graphiquement de jolies fractales.*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>Objectifs</b>	<b>4</b>
<b>IV</b>	<b>Consignes générales</b>	<b>5</b>
<b>V</b>	<b>Partie obligatoire</b>	<b>7</b>
<b>VI</b>	<b>Partie bonus</b>	<b>8</b>
<b>VII</b>	<b>Rendu et peer-évaluation</b>	<b>9</b>

# Chapitre I

## Préambule

Voici ce que Wikipédia a à dire sur la fracturation hydraulique :

La « fracturation hydraulique » est la dislocation ciblée de formations géologiques peu perméables par le moyen de l'injection sous très haute pression d'un fluide destiné à fissurer et micro-fissurer la roche. Cette fracturation peut être pratiquée à proximité de la surface, ou à grande profondeur (à plus de 1 km, voire à plus de 4 km dans le cas du gaz de schiste), et à partir de puits verticaux, inclinés ou horizontaux.

Cette technique relativement ancienne (1947), inventée pour les gisements d'hydrocarbures conventionnels, a vu son intérêt renouvelé par son association au forage horizontal (développé, lui, à partir de 1980). C'est la maîtrise graduelle de la rentabilité économique de cette association pour les gisements non-conventionnels, qui a guidé le développement récent de l'exploitation de ces derniers : elle a rendu accessibles des ressources autrefois soit inaccessibles, soit qui n'auraient été exploitables qu'à des coûts exorbitants et avec lenteur.

Elle est effectuée en fracturant la roche par un « stress » mécanique<sup>3</sup> à l'aide d'un fluide injecté sous haute pression à partir d'un forage de surface, pour en augmenter la macro porosité et moindrement la micro-porosité. Le fluide peut être de l'eau, une boue ou un fluide technique dont la viscosité a été ajustée.

Ce projet ne s'appelle pas *fract'oil* et par conséquent n'a aucun rapport avec la fracturation hydraulique.

# Chapitre II

## Introduction

Créé par Benoît Mandelbrot en 1974 à partir de la racine latine **fractus**, qui signifie "brisé" , "irrégulier", une figure fractale est un objet mathématique, telle une courbe ou une surface, dont la structure est invariante quelque soit le changement d'échelle.

De nombreux phénomènes naturels – comme l'aspect du chou romanesco – possèdent des formes fractales approximatives.



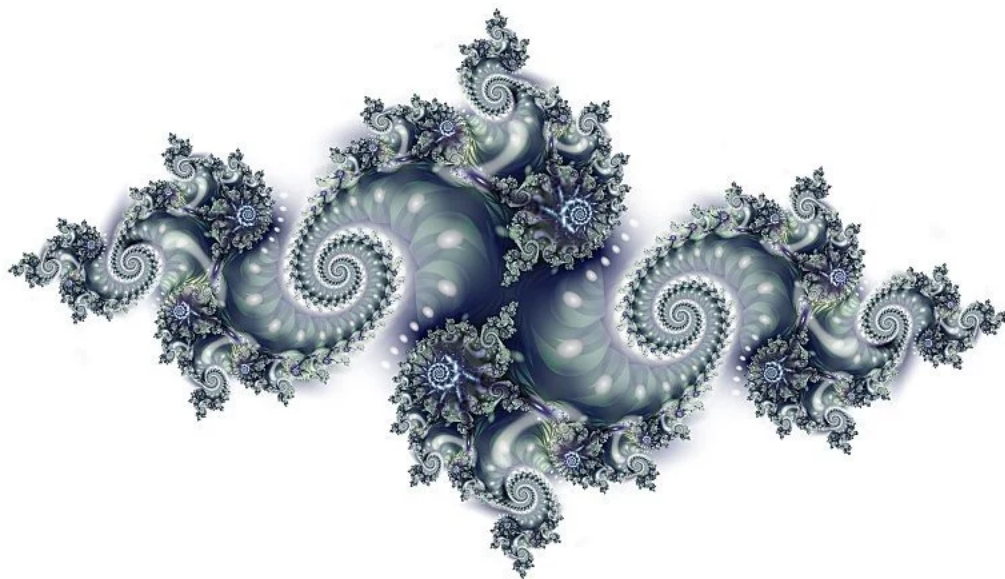
A vous aujourd'hui de vous essayer à cette pratique et de modéliser de magnifiques fractales !

# Chapitre III

## Objectifs

Maintenant que vous avez pris en main votre première bibliothèque graphique : la `miniLibX`, il est temps pour vous d'allumer tous les pixels de votre écran en même temps !

Ce nouveau projet sera pour vous l'occasion de vous perfectionner dans l'utilisation de la `miniLibX`, de vous faire découvrir/utiliser la notion mathématique des **nombres complexes** ainsi que de vous faire aborder la problématique de l'**optimisation** en programmation graphique.



N'oubliez pas de regarder les vidéos sur l'e-learning !

# Chapitre IV

## Consignes générales

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et de nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- L'exécutable doit s'appeler **fractol**.
- Vous devez rendre un **Makefile**.
- Votre **Makefile** devra compiler le projet, et doit contenir les règles habituelles. Il ne doit recompiler le programme qu'en cas de nécessité.
- Si vous êtes malin et que vous utilisez votre bibliothèque **libft** pour votre **fractol**, vous devez en copier les sources et le **Makefile** associé dans un dossier nommé **libft** qui devra être à la racine de votre dépôt de rendu. Votre **Makefile** devra compiler la librairie, en appelant son **Makefile**, puis compiler votre projet.
- Vous ne devez pas utiliser de variables globales.
- Votre projet doit être à la **Norme**.
- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (segmentation fault, bus error, floating point exception, etc...).
- Votre programme ne doit pas avoir de fuites mémoire.
- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier **auteur** contenant votre login suivi d'un '\n' :

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

- Vous devez **obligatoirement** utiliser la **miniLibX**. Soit dans sa version présente sur les dumps, soit à partir de ses sources. Si vous choisissez de travailler à partir de ses sources, vous devez appliquer les mêmes règles que pour votre **libft** telles que décrites au dessus.

- Dans le cadre de votre partie obligatoire, vous avez le droit d'utiliser les fonctions suivantes :
  - `open`, `read`, `write`, `close`
  - `malloc`, `free`
  - `perror`, `strerror`
  - `exit`
  - Toutes les fonctions de la lib `math` (`-lm` et `man 3 math`)
  - Toutes les fonctions de la `miniLibX`.
- Vous avez l'autorisation d'utiliser d'autres fonctions dans le cadre de vos bonus, à condition que leur utilisation soit dûment justifiée lors de votre évaluation. Soyez malins.
- Vous pouvez poser vos questions sur le forum, Slack, etc.

# Chapitre V

## Partie obligatoire

Ce projet consiste à créer un petit logiciel d'exploration fractale. Commencez donc par voir ce qu'est une fractale.

Les contraintes sont les suivantes :

- Votre logiciel doit proposer au minimum 3 types de fractales différentes, dont au moins l'ensemble de Julia et l'ensemble de Mandelbrot (soit au moins une troisième fractale au choix).
- Il doit être possible de faire varier avec la souris (sans clic) le paramètre de l'ensemble de Julia. Pour les autres types de fractale, cela est laissé à votre discrétion.
- La molette de la souris permet de zoomer et dézoomer, et cela de façon quasi infinie (modulo les limites de la machine). C'est le principe même des fractales.
- Il doit y avoir un jeu de couleur minimum pour ressentir la profondeur de chaque fractale. Mieux : lâchez-vous sur les effets psychédéliques.
- Un paramètre est passé en ligne de commande pour définir quel type de fractale est à afficher. Si il n'y a pas de paramètre fourni, ou si le paramètre est invalide, le programme affiche la liste des paramètres disponibles et quitte proprement.

En ce qui concerne la représentation graphique :

- La touche ESC permettra de quitter le programme.
- L'utilisation des `images` de la `minilibX` est fortement conseillée.

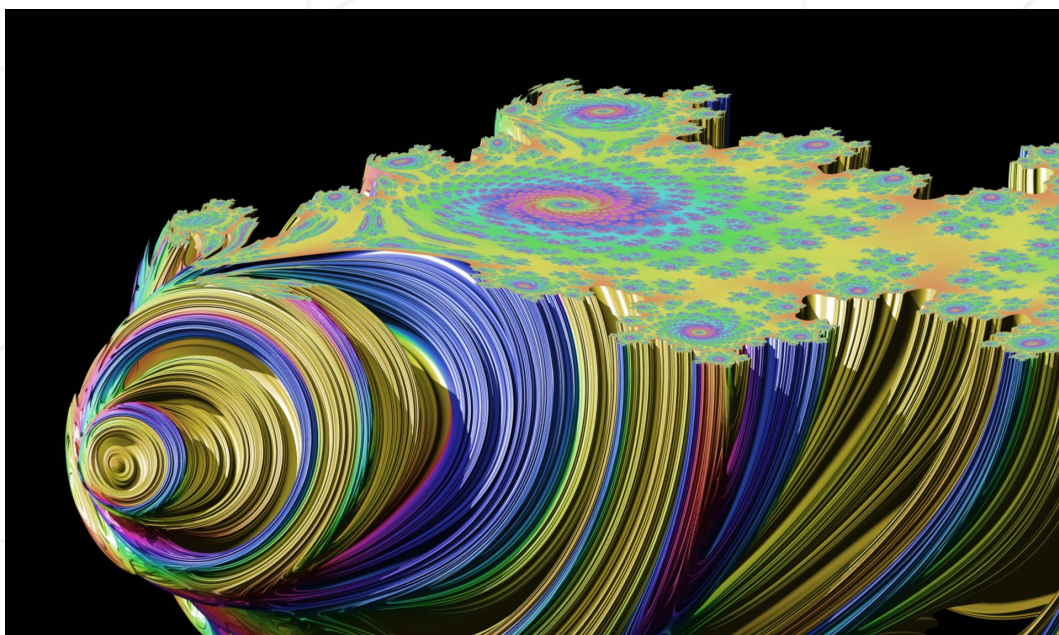


# Chapitre VI

## Partie bonus

Voici quelques idées de bonus intéressants à réaliser. Vous pouvez évidemment ajouter des bonus de votre invention, qui seront évalués à la discrétion de vos correcteurs.

- Le zoom pointe sur la position actuelle de la souris.
- En plus du zoom : déplacement avec les flèches.
- Faire tourner la palette de couleur.
- Implémenter le multi-threading.
- Plein de fractales sympa (des sites recensent plus d'une centaine de types différents).
- Représentations en 3D.
- Deux paramètres valides en ligne de commande, deux fenêtres pour deux fractales.



# Chapitre VII

## Rendu et peer-évaluation

Rendez-votre travail sur votre dépôt GiT comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance.

