



fract'ol

Art fractal

Résumé:

Ce projet consiste à créer graphiquement de jolies fractales.

Version: 2

Table des matières

I	Préambule	2
II	Introduction	3
III	Objectifs	4
IV	Règles communes	5
V	Partie obligatoire	6
V.1	Rendu	7
V.2	Gestion graphique	7
VI	Partie bonus	8
VII	Rendu et peer-evaluation	9

Chapitre I

Préambule

Voici ce que Wikipedia 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 » mécanique³ 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.



À vous aujourd'hui de vous essayer à cette pratique et de générer de magnifiques fractales !

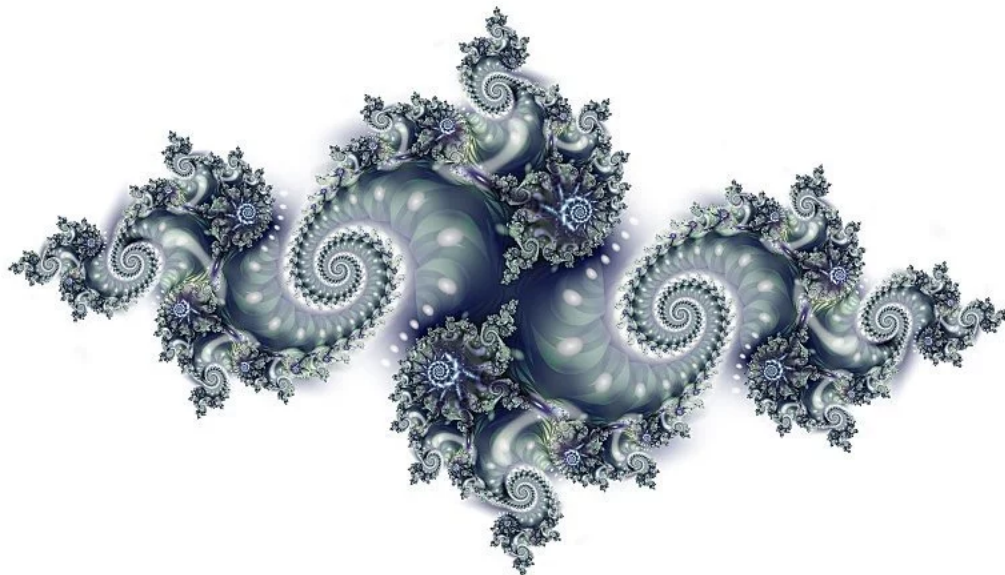
Chapitre III

Objectifs

Il est temps pour vous d'attaquer votre premier projet graphique !

Vous allez maintenant prendre en main la bibliothèque graphique de l'école : la **MiniLibX** ! Cette bibliothèque a été développée en interne et inclut des outils basiques permettant d'ouvrir une fenêtre, de créer des images et de gérer des événements clavier et souris.

Ce nouveau projet sera pour vous l'occasion de débiter dans l'utilisation de la **MiniLibX**, de découvrir ou d'utiliser la notion mathématique des **nombres complexes**, d'aborder la problématique de **l'optimisation** en programmation graphique, ainsi que la **gestion des événements**.



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

Chapitre IV

Règles communes

- Votre projet doit être écrit en C.
- Votre projet doit être codé à la Norme. Si vous avez des fichiers ou fonctions bonus, celles-ci seront incluses dans la vérification de la norme et vous aurez 0 au projet en cas de faute de norme.
- Vos fonctions ne doivent pas s'arrêter de manière inattendue (segmentation fault, bus error, double free, etc) mis à part dans le cas d'un comportement indéfini. Si cela arrive, votre projet sera considéré non fonctionnel et vous aurez 0 au projet.
- Toute mémoire allouée sur la heap doit être libérée lorsque c'est nécessaire. Aucun leak ne sera toléré.
- Si le projet le demande, vous devez rendre un Makefile qui compilera vos sources pour créer la sortie demandée, en utilisant les flags `-Wall`, `-Wextra` et `-Werror`. Votre Makefile ne doit pas relink.
- Si le projet demande un Makefile, votre Makefile doit au minimum contenir les règles `$(NAME)`, `all`, `clean`, `fclean` et `re`.
- Pour rendre des bonus, vous devez inclure une règle `bonus` à votre Makefile qui ajoutera les divers headers, bibliothèques ou fonctions qui ne sont pas autorisées dans la partie principale du projet. Les bonus doivent être dans un fichier différent : `_bonus.{c/h}`. L'évaluation de la partie obligatoire et de la partie bonus sont faites séparément.
- Si le projet autorise votre `libft`, vous devez copier ses sources et son Makefile associé dans un dossier `libft` contenu à la racine. Le Makefile de votre projet doit compiler la bibliothèque à l'aide de son Makefile, puis compiler le projet.
- Nous vous recommandons de créer des programmes de test pour votre projet, bien que ce travail **ne sera pas rendu ni noté**. Cela vous donnera une chance de tester facilement votre travail ainsi que celui de vos pairs.
- Vous devez rendre votre travail sur le git qui vous est assigné. Seul le travail déposé sur git sera évalué. Si Deepthought doit corriger votre travail, cela sera fait à la fin des peer-evaluations. Si une erreur se produit pendant l'évaluation Deepthought, celle-ci s'arrête.

Chapitre V

Partie obligatoire

Nom du programme	fractol
Fichiers de rendu	Makefile, *.h, *.c
Makefile	NAME, all, clean, fclean, re
Arguments	Le type de fractale à afficher et toute autre option disponible
Fonctions externes autorisées	<ul style="list-style-type: none">• open, close, read, write, malloc, free, perror, strerror, exit• Toutes les fonctions de la bibliothèque mathématique (option de compilation -lm, man man 3 math)• Toutes les fonctions de la MiniLibX• ft_printf et tout équivalent que VOUS avez codé
Libft autorisée	Oui
Description	Ce projet consiste à créer un petit logiciel d'exploration fractale. Commencez donc par voir ce qu'est une fractale.

Votre projet doit respecter les règles suivantes :

- Vous **devez** utiliser la MiniLibX. Soit la version disponible sur les machines de l'école, soit en l'installant par les sources.
- Vous devez rendre un **Makefile** qui compilera vos fichiers sources. Il ne doit pas **relink**.
- Les variables globales sont interdites.

V.1 Rendu

- Votre programme doit proposer les ensembles de **Julia** et **Mandelbrot**.
- La molette de la souris doit permettre de zoomer et dézoomer, et cela de façon quasi-infinie (dans les limites de la machine). C'est le principe même des fractales.
- En donnant des paramètres différents au programme, il doit être possible de créer différents ensembles de Julia.
- Au moins un paramètre est passé en ligne de commande pour définir quel type de fractale est à afficher dans la fenêtre.
 - Il est possible de gérer des paramètres en plus afin de les utiliser pour le rendu.
 - Si aucun paramètre n'est fourni ou si le paramètre est invalide, le programme doit afficher la liste des paramètres disponibles et quitter proprement.
- Il doit y avoir un **jeu de couleur** minimum pour ressentir la profondeur de chaque fractale. Mieux : lâchez-vous sur les effets psychédéliques.

V.2 Gestion graphique

- Votre programme doit afficher une image dans une fenêtre.
- La gestion de la fenêtre doit rester fluide (changer de fenêtre, la réduire, etc.).
- Appuyer sur la touche **ESC** doit fermer la fenêtre et quitter le programme proprement.
- Cliquer sur la croix en haut de la fenêtre doit fermer celle-ci et quitter le programme proprement.
- Utiliser les **images** de la MiniLibX est obligatoire.

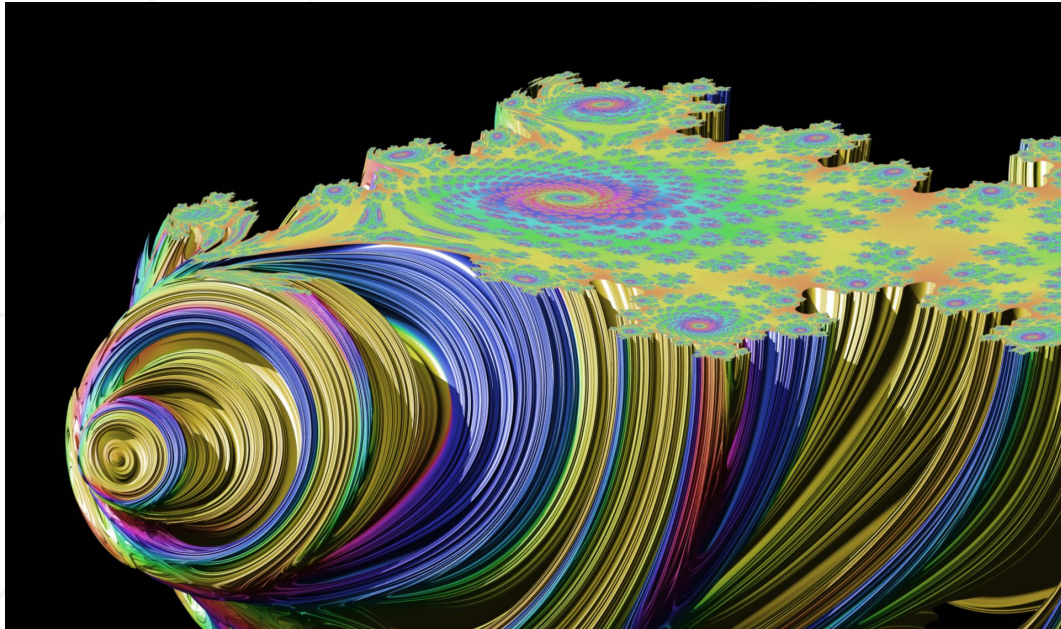
Chapitre VI

Partie bonus

Habituellement, vous seriez encouragé(e) à développer vos propres bonus. Cependant, d'autres projets graphiques plus intéressants sont à venir. Ne perdez pas de temps !

Vous aurez des points supplémentaires si :

- Il y a une fractale différente supplémentaire (il existe plus d'une centaine de types différents de fractales référencés en ligne).
- Le zoom suit la position de la souris.
- En plus du zoom : on peut déplacer la vue avec les touches flèches.
- Les couleurs changent (*color shift*).



Les bonus ne seront évalués que si la partie obligatoire est PARFAITE. Par parfaite, nous entendons complète et sans aucun dysfonctionnement. Si vous n'avez pas réussi TOUS les points de la partie obligatoire, votre partie bonus ne sera pas prise en compte.

Chapitre VII

Rendu et peer-evaluation

Rendez votre travail sur votre dépôt `Git` comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.

Vu que votre travail ne sera pas évalué par un programme, organisez vos fichiers comme bon vous semble du moment que vous rendez les fichiers obligatoires et respectez les consignes du sujet.

