

# Projet de programmation en C n°3 – L1 2018

## *Fractales*

© 2013 – 2014 – Franck Lepoivre –  $v_{0.2}$

21 février 2014

### Sujet

*Note : version alpha qui fera l'objet de mises à jour d'ici fin février. Les élèves qui ont de bonnes idées de fonctionnalités associées sont invités à les suggérer pendant la séance de TP du 19 février.*

Concevoir et réaliser en langage C un outil générateur d'images de fractales *a minima* spécialisé pour le traçage de l'ensemble de Mandelbrot, et si possible pour tracer tout type de fractale.

Pour en savoir plus :

- <http://fr.wikipedia.org/wiki/Fractale>
- [http://fr.wikipedia.org/wiki/Ensemble\\_de\\_Mandelbrot](http://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot)

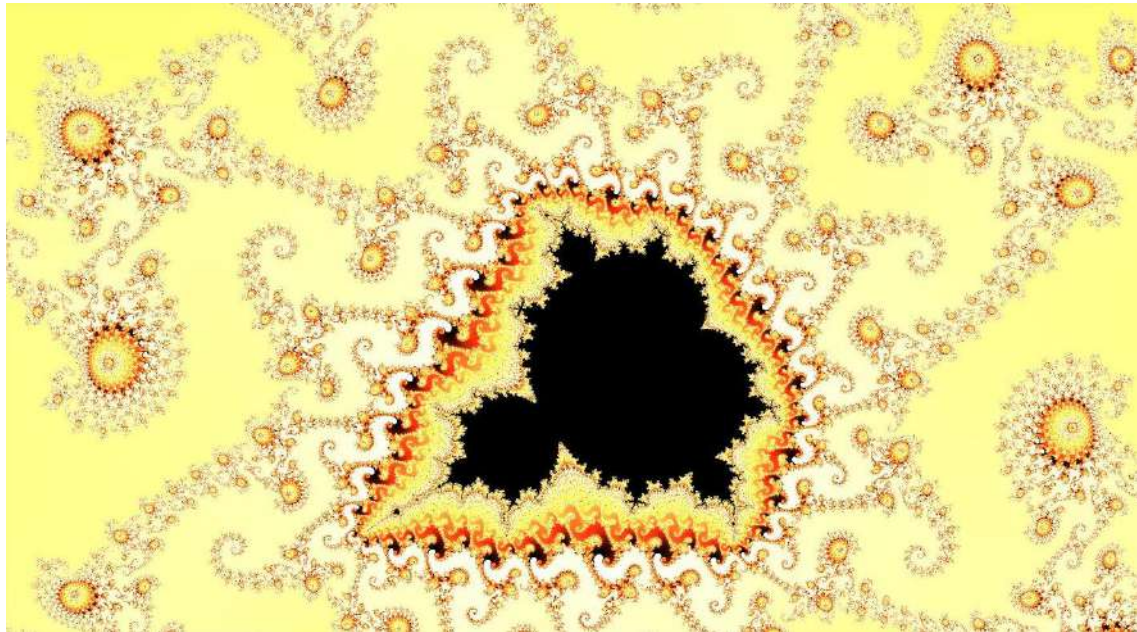


FIGURE 1 – Ensemble de Mandelbrot - vue d'ensemble

---

## Cahier des charges

### Cahier des charges impératif

Pour démarrer :

1. Concevoir et réaliser un couple `.c` et `.h` pour définir la structure et les opérations sur les nombres complexes.
2. Concevoir et réaliser une fonction pour tester si un nombre complexe en particulier est dans l'ensemble de Mandelbrot ou non.
3. Dessiner l'ensemble de Mandelbrot, dans le cadre  $(-2, 1)$  à  $(1, -1)$ , en noir et blanc, soit en console, soit vers un fichier PBM ([http://fr.wikipedia.org/wiki/Portable\\_pixmap](http://fr.wikipedia.org/wiki/Portable_pixmap))

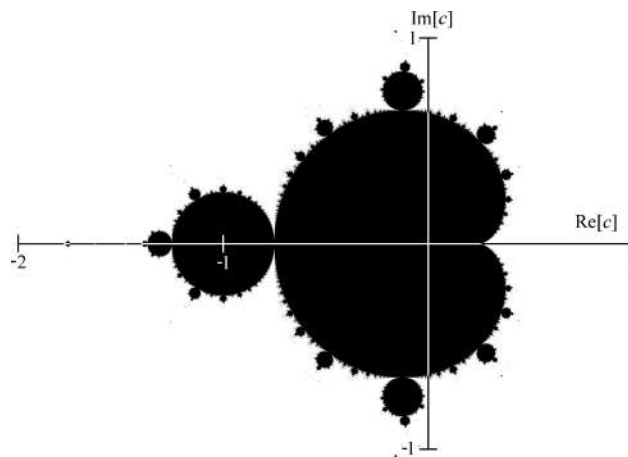


FIGURE 2 – Ensemble de Mandelbrot, dans le cadre  $(-2, 1)$  à  $(1, -1)$ , en noir et blanc

Pour poursuivre :

1. Affiner votre fonction pour quelle retourne un nombre d'itération soit avant convergence, soit avant divergence<sup>1</sup>.
2. Définir des fonctions qui permettent d'associer à un intervalle d'entiers une palette de couleurs (RGB).
3. Se servir de ces fonctions pour générer des versions colorées de l'ensemble de Mandelbrot. Vous pourrez par exemple utiliser des fichiers PGM et PPM pour visualiser le résultat.

---

1. Vous ne pouvez comprendre cet élément d'énoncé sans avoir étudié la page wikipedia à propos de l'ensemble de Mandelbrot

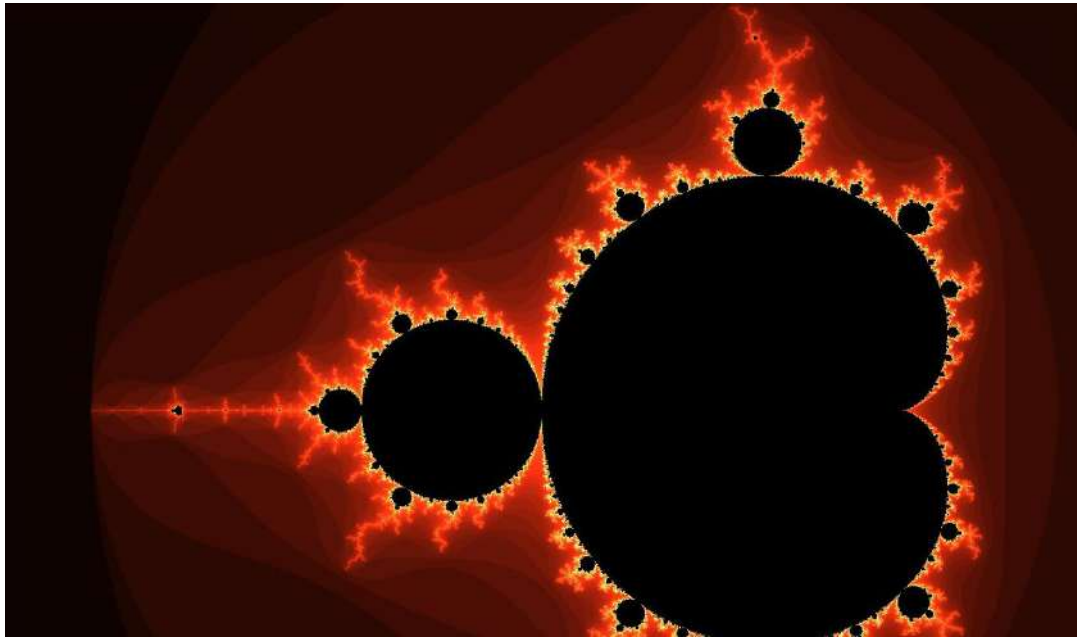


FIGURE 3 – Ensemble de Mandelbrot - vue d'ensemble

Fonctionnalités suivantes :

- Affichage à l'aide d'une librairie graphique.
- Zoomer sur un point donné, ou sur une sous-fenêtre donnée.
- Modifier interactivement la palette de couleur pour une image déjà tracée.
- Produire une image animée (GIF) du mouvement de zoom (modèle ici : [http://fr.wikipedia.org/wiki/Fichier:Mandelbrot\\_sequence\\_new.gif](http://fr.wikipedia.org/wiki/Fichier:Mandelbrot_sequence_new.gif)).

### **Axes pour se démarquer et exprimer ses talents (uniquement pour les plus motivés)**

Vous avez besoin de défis pour vous dépasser ?

Quelques axes pour vous démarquer :

- Design époustouflant.
- Chargement / sauvegarde des paramètres de traçage d'une fractale.
- Dialogue interactif ergonomique avec l'utilisateur pour l'aider à composer sa propre fractale.
- ...

---

## Consignes et informations pratiques

### Collaboration

Travaux de programmation, de rédaction de rapport et de soutenance individuels !

Le partage d'idées et la collaboration pour la conception sont autorisés.

### Livraison

Les consignes générales ci-dessous, formulées pour le cas d'un binôme, valent pour des équipes de toute taille, c'est-à-dire autant pour un monôme que pour un trinôme.

Remise au plus tard le 02/04/2014. Dès le lendemain, application d'une pénalité de 2 points par jour (au *pro rata temporis*). Attention, vérifiez que votre livrable a bien été reçu et ne vous a pas été retourné par GMail.

Une seule remise par équipe.

Une fois les notes publiées, un membre de l'équipe dont le nom aurait été omis tant dans l'objet de l'email de livraison, que dans le nom du livrable, et qui n'apparaîtrait pas dans le `README.txt`, ne pourrait prétendre à être rétabli dans sa note (elle serait définitivement 0). Pour vous assurer que la remise a été faite correctement par votre coéquipier, insistez pour être en copie de celle-ci.

Le livrable se présente sous la forme d'une archive ZIP de nom : `PROJET C L1 GP - NOM1 - NOM2.ZIP` envoyé par email à l'adresse de votre tuteur de projet (celui qui assure votre soutenance), ou par tout autre moyen de substitution qu'il vous indiquerait.

`NOM1` et `NOM2` sont vos noms respectifs, sans accent, d'où tirets ou autres blancs sont remplacés par le caractère '\_' et dont toutes les lettres sont capitalisées. Par exemple, si vous vous appelez respectivement Raymond de Sèze et Dexter Morgan et faisiez partie du groupe A, il faudrait rendre le fichier `PROJET C L1 A - DE_SEZE - MORGAN.ZIP`

Le non respect de l'une ou l'autre des consignes vaut pénalité.

### Objet PROJET C L1 GP - NOM1 - NOM2

Itheri Yahiaoui : `itheri.yahiaoui@inria.fr`

Franck Lepoivre : `franck.lepoivre@gmail.com`

Kaïs Klai : `kais.klai@lipn.univ-paris13.fr`

Kamel Chabchoub : `chabchoub.efrei@gmail.com`

Fabien Calcado : `fabien.calcado@gmail.com`

### Livrable

- Vos fichiers sources.
- SURTOUT PAS D'EXÉCUTABLE ni de fichiers `.o` ! (Sinon, cela entraîne un rejet de GMail - pour supprimer vos `.exe` et `.o` il suffit de supprimer les répertoires `bin` et `obj`)
- Votre fichier projet (idéalement CodeBlocks).
- Votre documentation (Rapport et autres documents) au format PDF.
- Un fichier `README.txt` qui comprend impérativement deux sections :
  - Un état d'avancement honnête et fiable de vos travaux en 10 lignes maximum.
  - Les références aux sources qui ont pu inspirer votre conception (la reprise de code est formellement interdite).
- Et en option, une mise en avant de fonctionnalités supplémentaires que vous aurez choisi de réaliser dans le but d'obtenir une meilleure note.

---

## Rapport

**Un sommaire** détaillé avec pages numérotées.

**L'analyse fonctionnelle générale** D'environ cinq pages, elle explique quelles sont les principales données traitées et quelles sont les principales fonctionnalités du programme, comment on peut les regrouper, comment elles sont organisées dans le programme. Cette analyse aboutit à la présentation des modules fonctionnels qui sont décrits en détail dans l'analyse détaillée. L'analyse générale doit préciser le ou les choix importants que vous avez faits pour la conception de votre projet.

**L'analyse fonctionnelle détaillée** D'environ dix pages, elle précise, pour chaque module, son utilité, ses entrées et ses sorties, les données qu'il modifie. Il faut décrire les données utilisées et les algorithmes mis au point (logique de manipulation) avec rigueur, concision et clarté, le tout accompagné de schémas. C'est un travail de synthèse, de pédagogie et de communication ; les extraits de code sont proscrits de vos rapports.

**Annexes** Elles présentent au moins les technologies tierces employées. Et comme option recommandée, bibliographie et webographie listent les références qui ont permis d'étayer votre analyse et votre conception.

## Évaluation

### Barème

**10 pts** pour votre programme, notamment en regard de la qualité et de l'extension des fonctionnalités qu'il offre.

**10 pts** pour votre communication :

**5 pts** en fonction de la qualité de votre communication orale (soutenance),

**5 pts** en fonction de la qualité de votre communication écrite (rapport).

### Quelques conseils

Il faut que votre communication orale soit claire, concise, efficace.

Il faut que votre programme :

- compile et s'exécute (sinon vous ne serez noté que sur 5),
- fonctionne conformément aux attentes (fonctionnalités de base + options éventuelles),
- soit écrit proprement (conventions d'écriture) et efficacement (algorithmique).

## Planning du projet

**Distribution du sujet de projet** 19/02/2014 (version initiale).

**TP de lancement de projet** 19/03/2014 (sujet en version finale).

**Soutenance de projet** 02/04/2014.