

# Projet de développement logiciel

## Réalisation d'un visualiseur de l'ensemble de Mandelbrot

### Objectif

Ce projet de programmation avancée a pour ambition de vous placer dans une situation concrète d'un développement logiciel en binôme, dans lequel vous êtes en charge du développement complet.

Un soin particulier devra être apporté à votre programme, sa documentation, sa maintenance et son évolutivité. Pour cela, vous mettrez en oeuvre les notions, outils et bonnes pratiques vus en cours de programmation avancée : makefile, compilation séparée, débogueur, SVN, ...

### 1 Plantons le décor ! - Fonctionnalités demandées

La page Wikipédia [http://fr.wikipedia.org/wiki/Ensemble\\_de\\_Mandelbrot](http://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot) décrit très bien ce qu'est l'ensemble de Mandelbrot. Le but de ce TP est de fournir un logiciel de visualisation graphique de cet ensemble, qui permet de l'admirer et de zoomer sur les parties intéressantes.

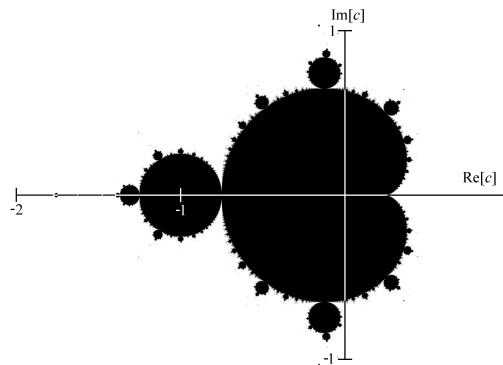


FIGURE 1 – L'ensemble de Mandelbrot - Source Wikipédia

### 2 Figures Imposées

Pour réaliser votre tâche, vous allez vous appuyer sur :

- deux bibliothèques existantes : GSL (calcul scientifique) et GL (visualisation graphique) ;
- du code existant écrit par un développeur tiers. Le code correspond à un ensemble de fonctions qui vous seront utiles pour afficher une fenêtre 2D dans laquelle vous pourrez afficher des points ;

- un dépôt SVN avec accès restreint pour votre binôme sur lequel vous pourrez déposer votre projet et faire les opérations usuelles (commit, update).

Voici les différentes versions que l'on vous demande de traiter :

- **Version 1** : dans une fenêtre graphique de taille fixée à l'avance, on demande d'afficher les (des!) points de l'ensemble en blanc sur fond noir. Toutes les allocations pourront être statiques, et les constantes fixées à l'avance. La seule contrainte est de faire les calculs sur des nombres complexes **GSL**.
- **Version 2** : toujours en affichant l'ensemble entier, l'utilisateur peut choisir la précision des calculs. On utilisera des couleurs différentes suivant le nombre d'itérations nécessaires pour décider si un point est dans l'ensemble ou non (comme sur la figure 2).

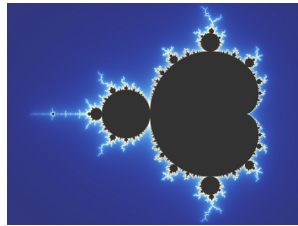


FIGURE 2 – L'ensemble de Mandelbrot coloré - Source Wikipédia

- **Version 3** (optionnelle) : l'utilisateur peut zoomer sur une partie de l'image. Il faudra donc adapter au zoom la précision des calculs, et éventuellement la taille de la fenêtre.

## 2.1 Séance 1 - Prise en Main, Makefile, SVN, et début de développement

**Préliminaires** Avant de commencer, chaque binôme va récupérer le source du sujet (quelques fichiers sources et autres choses utiles) :

1. Récupérer le sujet entier (avec les sources) sur le SVN de l'école :<sup>1</sup> :

```
svn checkout svn+ssh://synthe/ima3/sujetPA2011
```

2. **À un autre endroit dans votre arborescence**, récupérer votre dépôt SVN à l'aide de la liste des binômes (présente dans l'archive). Chaque binôme a un dépôt SVN personnel, et seules les personnes du binôme ont accès à ce dépôt.
3. À partir de ce moment là, vous ne travaillez que dans ce répertoire. Les changements effectués dans ce répertoire seront "sauvegardés" à l'aide de la commande `svn commit` (voir le cours!)

*REMARQUE 1 Pour travailler chez vous, il faudra faire en sorte de récupérer votre dépôt SVN de l'extérieur (c'est expliqué sur la page du cours), et il faudra éventuellement installer les bibliothèques **GSL** et **GL**.*

**Makefile et utilisation des bibliothèques** Dans la première séance, on vous demande de tester les bibliothèques. Pour cela, on vous demande de travailler dans votre copie locale, répertoire `Codes/version0/` :

- Tester **GSL** à l'aide d'un main : utiliser la bibliothèque pour faire la somme de deux complexes.
- Tester la bibliothèque graphique et les fonctionnalités fournies dans `display2D.c` et `util.c` en dessinant des carrés bleus dans la fenêtre graphique.

---

1. De l'extérieur, c'est plus compliqué, mais expliqué sur la page du cours

Faire un Makefile qui permet de compiler et linker correctement vos binaires utilisant ces bibliothèques. **Ne pas oublier d'ajouter avec svn add vos sources, ainsi que le Makefile, et de "commiter"**

Et ensuite, c'est à vous ! Pour la première version, vous pouvez vous inspirer de l'avancée proposée sur le sujet situé à l'URL<sup>2</sup> [http://www.point-libre.org/~ludo/drupal/sites/default/files/enseignement/C/gsl\\_td6.pdf](http://www.point-libre.org/~ludo/drupal/sites/default/files/enseignement/C/gsl_td6.pdf). Attention, notre avancée diffère à partir du point 7 (pas de sauvegarde dans un fichier), et on vous impose d'utiliser les bibliothèques GSL (complexes !) et GL.

## 2.2 La suite !

Pour la deuxième séance, changez de compte ! Pour travailler, il faudra donc faire un checkout initial si cela n'a pas été fait. Ensuite, on vous laisse libre, mais ne partez pas dans tous les sens.

**Les "mouvements" sur le SVN seront évalués. Il est donc important de bien travailler en binôme, en changeant de session pendant les TPs.**

## 3 Mise en oeuvre et consignes pour le rendu

**Mise en oeuvre** Expliquée plus haut (séance 1).

**Rendu** Nous récupérerons **le 6 Juin 2011 20h** (5 points par jour de retard) vos projets dans vos dépôts qui devront avoir la structure suivante :

- un fichier `Readme.txt` contiendra une description rapide de votre logiciel, de ses fonctionnalités et un mode d'emploi succinct.
- un répertoire `Code` (avec Makefile). Pour faciliter la correction, le binaire s'appellera `mandelbrot`. Chaque version sera dans un sous-répertoire à part.
- un répertoire `Tests` qui comprendra quelques scripts de tests.
- éventuellement, un répertoire `old` contenant du code inutile.
- un fichier `nomdubinome.pdf` contiendra votre rapport. Le rapport ne comprendra pas plus de 5 pages, devra être clair et précis et notamment comporter les limitations de votre outil.

**PAS de rapport papier SVP !**

**Attention ! votre dépôt SVN devra être propre, ie ne pas comporter de fichier .o, tilde, binaire, etc.**

Nous fournirons un script python qui permettra de vérifier les consignes. Des points seront enlevés aux binômes pour lesquels le script renvoie `Fail`.

**Modalités d'évaluation** Nous évaluerons la maîtrise des outils présentés lors du cours de Programmation Avancée, ainsi que la qualité de votre développement et de votre programme :

- les fonctionnalités, évidemment, mais aussi ...
- l'utilisation des bibliothèques ;
- le découpage des fonctions, les commentaires, le découpage en modules, l'arborescence du SVN ;
- les aspects maintenance (la doc programmeur, la lisibilité du code) ;
- les aspects utilisateur (la doc utilisateur, les exemples, ...)
- **la gestion de ces différents points durant les séances de TP sera aussi évaluée**

Évidemment, cette liste n'est pas exhaustive !

---

2. sous Licence Creative Commons