

1 Description

Le but de ce projet est de fabriquer un logiciel de visualisation des fractales de Mandelbrot, avec un calcul distribué.

Une fractale de Mandelbrot(https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot) est un ensemble défini sur le plan des réels et se représente donc facilement comme une image. En zoomant sur une partie de l'ensemble, on observe des structures très complexes.

Le projet doit présenter à l'utilisateur une interface web similaire à celle-ci <https://dev.windows.com/en-us/microsoft-edge/testdrive/demos/mandelbrot/> où il pourra sélectionner un rectangle du plan et un *nombre d'itérations*, le logiciel calculera alors la représentation de la fractale dans ce rectangle.

Le projet comporte deux parties, un serveur et des clients de calculs.

1. Le serveur a deux rôles :
 - (a) Il fournit un serveur web minimaliste qui affiche une page web à l'utilisateur dans laquelle il fait une requête pour visualiser une partie de la fractale. Une fois la fractale calculée, il renvoie l'image à l'utilisateur sous forme de page web.
 - (b) Pour chaque requête d'un utilisateur, il découpe la requête en plusieurs morceaux et envoie ces sous-requêtes à des clients de calculs. Une fois toutes les sous-requêtes calculées, le serveur assemble tous les résultats pour produire une image.
2. Les clients de calcul se connectent à un serveur à leur création puis attendent des requêtes de celui-ci. Lorsqu'une ou plusieurs requêtes sont reçues ils les calculent et renvoient le résultat au serveur.

2 Consigne

Le but de ce Projet est de construire une architecture réseau pour remplir une tâche complexe. Votre attention doit donc se tourner vers les aspects réseau, vous devez implémenter vous-même :

- Le serveur HTTP qui affiche l'interface utilisateur,
- La gestion des sérialisations entre vos clients et votre serveur,
- Le découpage de la tâche par le serveur.

Par contre vous pouvez utiliser une bibliothèque pour générer les images (ex ImageIO) par le serveur.

3 Interface utilisateur

La page web qui sert d'interface est disponible sur Eprel (ProjetPW.html). Votre serveur doit servir ce fichier aux utilisateurs qui s'y connecte en modifiant l'image pour qu'elle corresponde à celle demandée. Lisez le code source du début de la page.

4 Dates de soumission

4.1 Extension

Pour améliorer le projet, vous pouvez modifier les clients pour qu'ils utilisent des threads concurrents afin de tirer avantage de tous les processeurs de la machine sur lequel ils sont lancés

5 Définition de l'ensemble de Mandelbrot

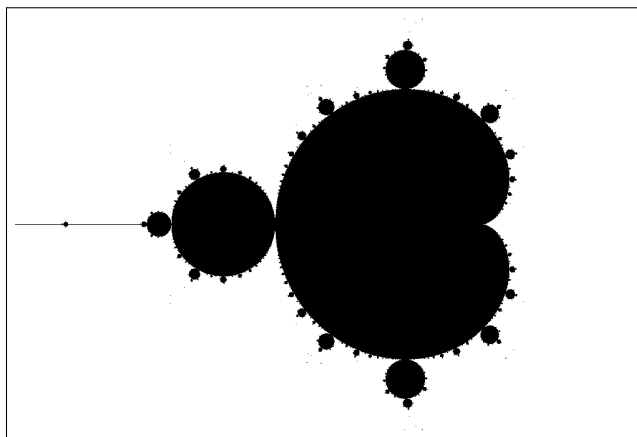
L'ensemble de Mandelbrot est défini à l'aide la suite complexe suivante

$$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$$

Le point complexe c appartient la n^{iem} itération de l'ensemble de Mandelbrot M_n si

$$|z_{n+1}| \leq 2$$

. Le dessin de l'ensemble en représentant les points appartenant à l'ensemble en noir et les autres en blanc produit l'image suivante :



Ici le cadre de l'image représente le rectangle de coin inférieur gauche $(-2,-1)$ et de coin supérieur droit $(1,1)$.

On pourra utiliser le code suivant pour tester l'appartenance du point (xc,yc) à l'ensemble de Mandelbrot M_n . La fonction renvoie n si le point appartient à l'ensemble, et un nombre strictement inférieur à n sinon.

```
public static int testMandelbrot(double xc, double yc, int n) {
    double xz = 0;
    double yz = 0;
    int i = 0;
    while( xz*xz + yz*yz <= 2*2 && i < n ) {
        double txz = xz;
        xz = xz*xz - yz*yz + xc;
        yz = 2.0*txz*yz + yc;
        i++;
    }
    return i;
}
```

Étant donné un rectangle à calculer (par exemple $(-2,-1)$ à $(1,1)$) et la taille de l'image à obtenir (par exemple 300×200). On fabrique l'image en calculant pour chacun des pixels de l'image ses coordonnées dans le rectangle (par exemple le pixel à la position (xp,yp) a pour coordonnées $(3*xp/300 - 2, 2*yp/200 - 1)$).