

Programmation et Conception Objet

Semestre 3, TP 6 : dessin de fractales

Durée : 2 TP (3 heures)

Samuel Delepoulle, Franck Vandewiele, Benjamin Danon

8 décembre 2015

Objectif : Le but du TP est d'afficher l'ensemble de Mandelbrot qui est une fractale.

Introduction

On s'intéresse à la suite complexe z_n définie par

$$z_{n+1} = z_n^2 + c \text{ avec } z_0 = 0$$

Cette suite définit l'ensemble de Mandelbrot. Dans le plan complexe, on représente en noir (ou dans une couleur quelconque) les points pour lesquels cette suite ne tend pas vers l'infini. On représente généralement en couleur les points pour lesquels elle diverge (on peut choisir une couleur qui est fonction de sa rapidité à diverger).

Classe pour représenter les nombres complexes

Ecrire une classe nommée **Complexe** qui permet de représenter un nombre complexe avec sa partie réelle et sa partie imaginaire (deux **double**).

Vous écrirez ensuite :

- les constructeurs nécessaires (au moins un constructeur avec paramètres) et la méthode `toString()` ;
- une méthode pour ajouter un nombre complexe à l'instance courante¹ ;
- une méthode pour multiplier l'instance courante par un autre nombre complexe² ;
- une méthode pour calculer le module du nombre complexe³
- vous pouvez ajouter une méthode qui calcule le carré du nombre complexe.

1. on rappelle que $z1 + z2 = (a + bi) + (c + di) = (a + c) + (b + d)i$

2. de même $z1.z2 = (a + bi)(c + di) = (ac - bd) + (ad + bc)i$

3. le module est défini comme $|(a + bi)| = \sqrt{a^2 + b^2}$

Classe Mandelbrot

Calcul de la suite

Ecrivez une nouvelle classe `Mandelbrot` qui permet :

1. de calculer la valeur suivante de la suite de Mandelbrot connaissant z_{n-1} et c .
2. de vérifier si la suite tend vers l'infini pour une valeur de c donnée. Ecrivez une méthode qui retourne un nombre entier (`int`). Il est démontré que dès que la valeur du module d'une des valeurs de la suite vaut 2 ou plus, la suite tend vers l'infini. Dans la pratique on calcule un certain nombre de termes jusqu'à une borne déterminée (par exemple 200). Si la valeur 2 est dépassée, on sait que la suite diverge et on retourne le nombre de pas effectués sinon, on considère que la suite converge⁴ et on retourne -1 ;

Dessin

Ecrivez la méthode :

`BufferedImage dessine(double r1, double i1, double r2, double i2, int larg, int haut)`

Elle retourne un `BufferedImage` qui représente l'ensemble de Mandelbrot dans le plan complexe (défini par quatre valeurs réelles). On précise la hauteur et la largeur de l'image générée.

Consultez la documentation de `BufferedImage` qui permet de gérer une image en mode point. Utilisez le constructeur qui permet de créer une image dans un mode prédéfini en RVB (ce qui correspond au type `BufferedImage.TYPE_INT_RGB`).

Vous utiliserez la méthode `setRGB(i,j,rgb)` qui permet de fixer la couleur d'un point. `rgb` étant un entier qui code une couleur. On peut le calculer en fonction des quantités rouge, vert, bleu et alpha (transparence) codées de 0 à 255 comme suit :

```
int rgb = (alpha<<24)+(rouge<<16)+(vert<<8)+bleu;
```

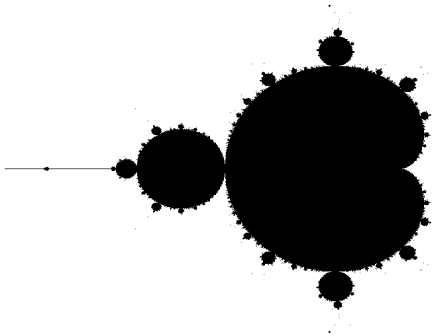
On peut aussi utiliser la méthode `java.awt.Color.HSBtoRGB` qui retourne un entier `rvb` à partir d'une teinte, d'une saturation et d'une valeur de brillance.

Enfin, un `BufferedImage` peut soit être affiché (il se comporte comme une `Image`) ou bien être sauvegardé dans un fichier :

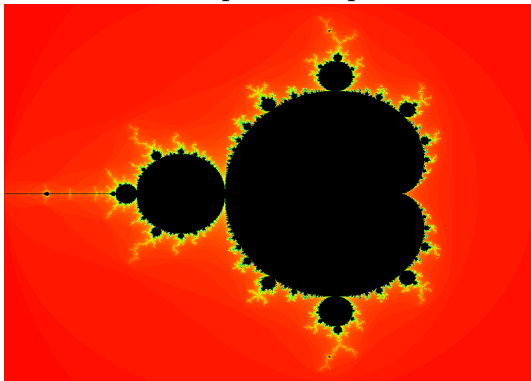
```
File file = new File("result.png");
try {
    ImageIO.write(im, "png", file);
} catch (IOException e) {
    System.out.println("Write error for " + file.getPath() +
                       ": " + e.getMessage());
}
```

4. on n'a pas de garantie que la suite converge pour ce nombre mais il est un bon candidat pour appartenir à l'ensemble de Mandelbrot

Dans un premier temps, parcourez les points, si le pixel appartient à l'ensemble de Mandelbrot, colorez le en noir et en blanc sinon. Vous pouvez créer une image de 800 par 600 pixels pour des valeurs réelles et imaginaires comprises entre -2 et $+2$. Sauvegardez le résultat dans un fichier pour visualiser le résultat. Si tout est correct, vous devez avoir une image qui ressemble à celle-ci :



Ajoutez ensuite la couleur en fonction du nombre de pas pour que le module dépasse 2. Vous obtiendrez par exemple ceci :



Améliorations

Affichez le `BufferedImage` dans une application (Un `JFrame`). Il doit être possible de zoomer sur une zone en cliquant avec la souris (et de "dézoomer" avec le bouton droit par exemple).