

Présentation du projet

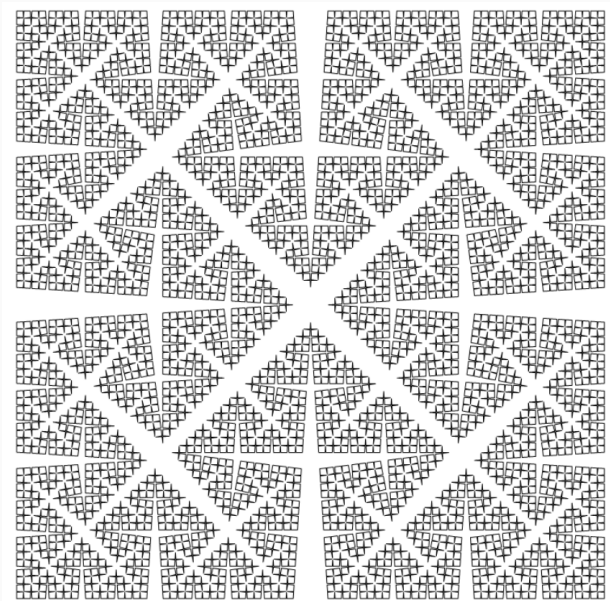
Dessiner des fractales

SINF1252

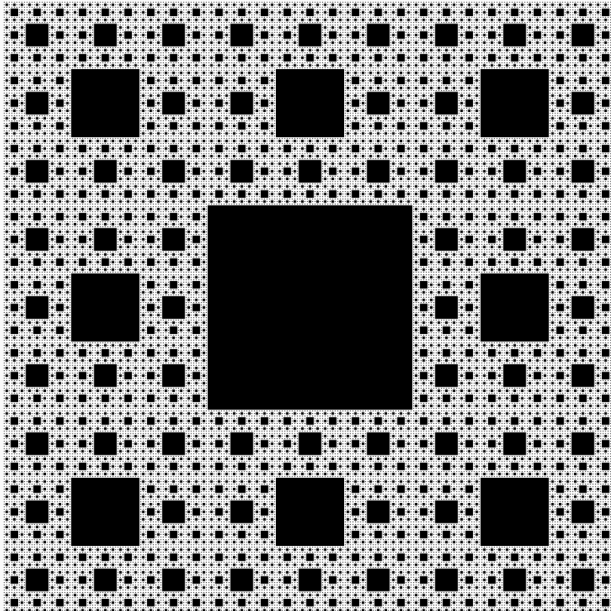
13/03/2018

- Objet mathématique
- Structure invariable à toute échelle
- Existe dans la nature
- Plusieurs types

Fractale Cesarò



Tapis de Sierpiński



Chou romanesco



$$z_{n+1} = z_n^2 + c$$

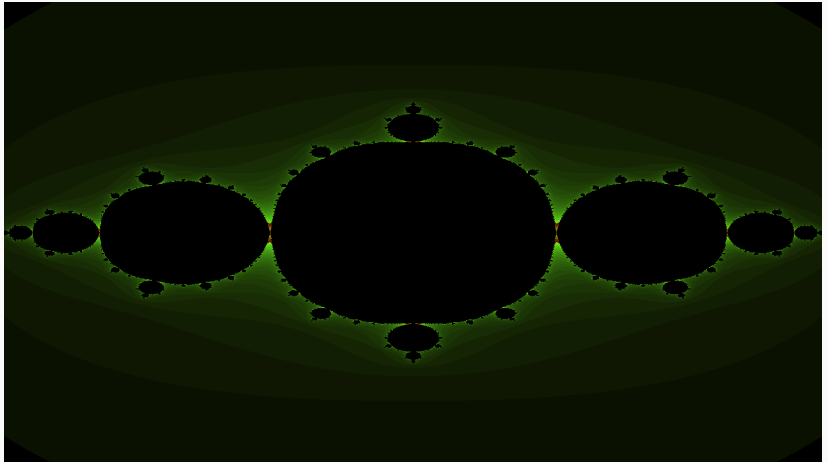
$$z, c \in \mathbb{C}$$

- Relation de récurrence, plan complexe $[-1; 1] \times [-1; 1]$
- $\forall c, c = a + bi, a \in [-1; 1], b \in [-1; 1], \exists$ une fractale

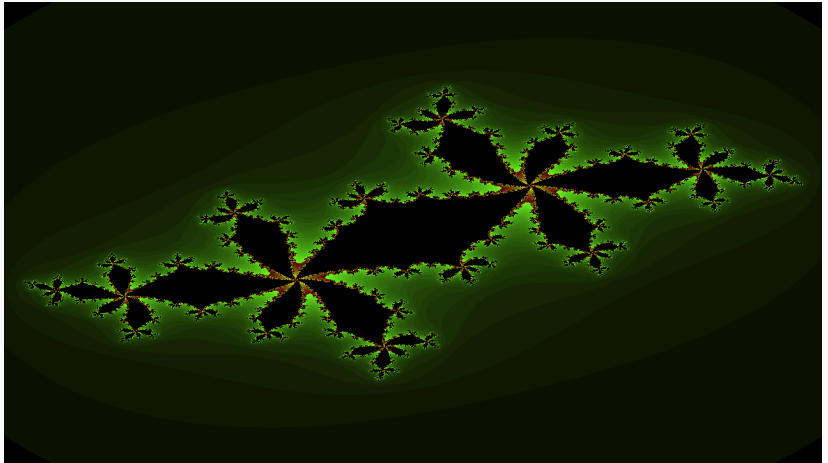
Mettre en image

- Fractale \rightarrow image de $N \times M$ pixels
- Associer chaque pixel (x, y) à un point $\in [-1; 1] \times [-1; 1]$
 - Nombre z
- Calculer la relation de récurrence pour chaque z , pour $c = a + bi$ donné
- Valeur du pixel = nombre d'itérations
 - Jusqu'à une condition d'arrêt
- Transformation nombre d'itérations \rightarrow couleur

Example: $c = -0.8 + 0.0i$



Example: $c = -0.52 + 0.57i$



- **Input:** un ou plusieurs fichiers, et/ou stdin
 - une description de fractale par ligne
 - nom, valeur $a + bi$, taille de l'image à générer
 - 1, 10, 1000, ..., N fractales par fichier
- **Output:** fichier bitmap (.bmp) représentant la fractale dont la moyenne arithmétique des valeurs des pixels est la plus haute

$$\max \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} V(x, y) \right)$$

En pratique: calculer une fractale

```
int w, h, x, y;  
w = fractal_get_width(f);  
h = fractal_get_height(f);  
for (x = 0; x < w; x++) {  
    for (y = 0; y < h; y++) {  
        fractal_compute_value(f, x, y);  
    }  
}
```

Architecture (S8)

- Calcul d'une fractal coûteux, surtout avec de grandes images
 - Full HD: $1920 \times 1080 = 2,073,600$ pixels
- Tâche parallélisable
 - → Multi-threading (bibliothèque pthread)
- Concrètement,
 - Quelles sont les structures de données utilisées?
 - Quels sont les types de threads présents (càd, que font-ils)?
 - Comment vos threads vont-ils communiquer?
 - Que communiquent vos threads?
- Phase d'architecture, **PAS de code**
- Formatif

- Un template de code + une API vous seront fournis
 - `fractal_compute_value` donné
 - Applique relation de récurrence sur un pixel donné
 - `write_bitmap_sdl` donné
 - Transforme une fractale en image bitmap
- **Tests unitaires**
- Utilisation de `git`

- Juste après la phase d'implémentation
- Réception de 5 projets
 - Reviews de 2 d'entre-eux
- Phase **individuelle**
- Plus de détails le moment venu

- S7: Enregistrement des groupes sur Moodle/Inginious
- S8: Interviews d'architecture
- S11: permanences
- S12: remise du projet
- S13: remise des reviews