

## Mini projet FMJ – Outils de programmation

---

L'ensemble de Mandelbrot est défini dans le plan complexe. Pour tout complexe  $c$ , on considère la série  $z_{i+1} = z_i^2 + c$  avec  $z_0 = 0$  et on calcule le nombre d'itérations requis pour obtenir  $|z_i| > 2$ . Pour certaines valeurs de  $c$  (qui forment l'ensemble de Mandelbrot), cela n'arrive jamais. De ce fait, il faut imposer un nombre maximal d'itérations (par exemple 1000).

Il est d'usage de représenter l'ensemble de Mandelbrot sous la forme d'une image, qui est fractale. Chaque pixel  $(x, y)$  de l'image correspond à un complexe  $c = x + y.i$ , et la couleur du pixel est déterminée par le nombre d'itérations requises pour obtenir  $|z_i| > 2$  (en utilisant une palette de couleurs).

Les ensembles de Julia sont définis de manière similaire. Un ensemble de Julia correspond à une constante  $c$  complexe. Pour chaque point  $(x, y)$  de l'image, on fixe  $z_0 = x + y.i$  puis on compte le nombre d'itérations de  $z_{i+1} = z_i^2 + c$  pour obtenir  $|z_i| > 2$ . De nouveau, le nombre d'itérations détermine la couleur du point.

Vous devez écrire un programme qui va générer une animation représentant une succession d'images des ensembles de Mandelbrot, de Julia, ou d'autres ensembles qui peuvent se définir de manière similaire. Chaque image de votre animation pourra correspondre à un zoom progressif sur une partie de l'image, ou à l'utilisation de paramètres différents (par exemple la constante  $c$  pour l'ensemble de Julia).

Votre programme devra utiliser toutes les ressources disponibles pour calculer ces images avec un niveau maximal de parallélisme. En particulier, il faut utiliser tous les coeurs disponibles, et également le GPU s'il est disponible.

Vous pouvez générer des images au format PPM (ou un autre format si vous préférez) et générer ensuite une vidéo avec par exemple, la command `"ffmpeg -y -r 24 -i f%06d.png -an -vcodec h264 video.mp4"`. Vous pouvez aussi afficher les images à l'écran en temps "réel".

Un fichier de configuration doit permettre de choisir la fractale à générer, la taille des images et de décrire comment les paramètres de cette fractale doivent évoluer au cours du temps.

Votre programme doit gérer au moins les ensembles de Mandelbrot et de Julia, mais doit aussi pouvoir être facilement adaptable à toute fractale où la couleur d'un point du plan est calculée par une fonction.

Vous devez écrire votre programme en C++, en utilisant les bonnes techniques de programmation. En particulier, il est vivement conseillé :

- de définir des classes,
- d'utiliser le polymorphisme,
- d'associer à un nom de fractale à générer la fonction qui permet de la calculer, ainsi que les paramètres qu'elle accepte,
- d'utiliser au mieux la programmation parallèle (CPU et GPU)

Vous avez la possibilité d'ajouter des extensions à ce projet.