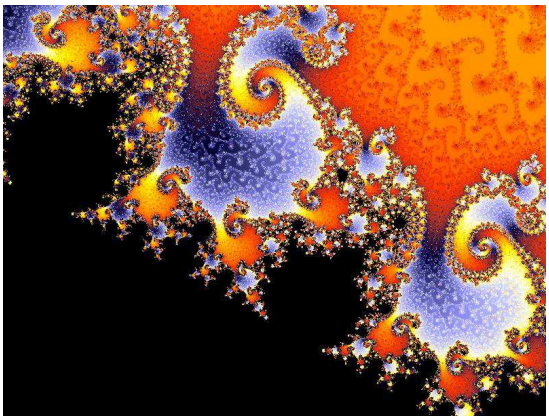
Conception et Réalisation de Logiciels – Phelma

Projet : FRACTALES

Thème du projet :

Il s'agit de développer un générateur de fractales en s'inspirant de l'applet suivante :
http://www.crocodilus.org/jeux/fractales/fractales.htm



Objectifs du projet :

- Calcul et affichage des 4 fonctions fractales suivantes :
  - F(z) = z2 + c
  - F(z) = cos(z) x c
  - F(z) = sin(z) x z0
  - F(z) = z2 + z0
- Le type C99 complex ainsi que la librairie « complex » des fonctions de calculs sur les nombres complexes seront utilisées (creal, cimag, csin, etc…
- Paramètres modifiables par l'interface utilisateur :
  - Point BasGauche (plan complexe)
  - Point HautDroit (plan complexe)
  - Module de sortie
  - Profondeur
  - Constante c
  - Choix de la fonction fractale à utiliser
- Gestion des couleurs :
  - Gestion d'un maximum de 8 couleurs primaires modifiables dans l'interface
  - Calcul de dégradés de couleur entre ces couleurs primaires (cf. annexe)
- Fonctions « enregistrer » et « charger » les paramètres de la fractale (fonction + variables) afin de garder des configurations remarquables
- Export du fichier image calculé au format PPM (cf. annexe)
- Fonction « Quitter »

Extensions possibles :

- Implémenter d'autres fonctions complexes F(Z) (cf. http://www.crocodilus.org/jeux/fractales/fractales.htm)
- Fonction « Zoom »

Difficultés principales du projet :

- Compréhension du sujet.
- Gestion du zoom.

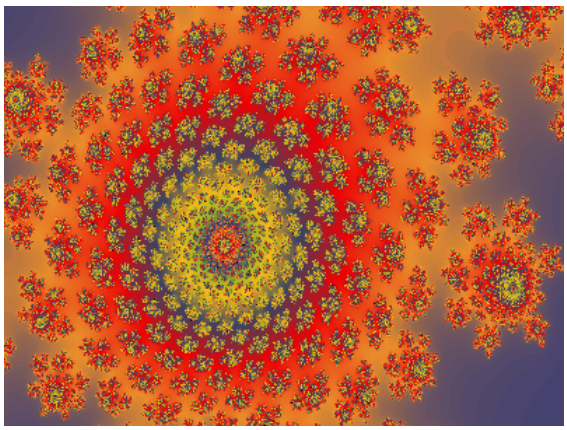
Annexes

Principes de calcul des images fractales et de la palette de couleurs

- On définit un plan complexe, associant à chaque point (I,J) de l'image un complexe Z_0.
- Une image fractale est obtenue à partir d'une suite de nombres complexes : $Z_{i+1} = F(Zi)$
- Pour chaque point I,J de l'image, on prend le Z_0 associé dans un plan (rectangle) complexe donné.
- On calcule ensuite la série jusqu'à ce que | Zn | atteigne une valeur de sortie choisie ou que n atteigne une profondeur donnée.
- Si la profondeur a été atteinte (la suite diverge) le point I,J ne fait pas partie de la forme fractale
- Si la profondeur n'a pas été atteinte (la suite converge) le point I,J fait partie de la forme fractale
- Pour colorer l'image, on affiche pour chaque point une couleur associée à la profondeur atteinte

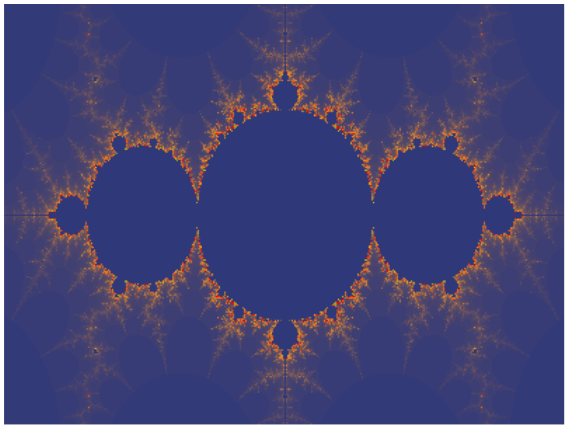
On n'obtient pas de forme fractale pour n'importe quels F, plan complexe, valeur de sortie, profondeur. Les fractales de type Julia ont une fonction F(z) qui inclut une constante complexe c. Les fractales de type Mandelbrot ont une fonction F(z) où z0 fait office de constante.

Exemples :



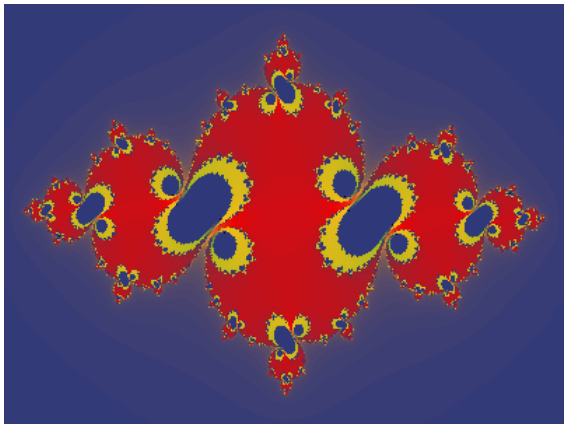
L'image ci contre (type Julia) est obtenue avec :

- $F(z) = z^2 + c$ avec c = -0.39492 + 0.59568 i
- Plan complexe défini par les points bas-gauche (-0.75, 0.6) et haut-droit (0.21, 0)
- Module de sortie : 4
- Profondeur : 1000



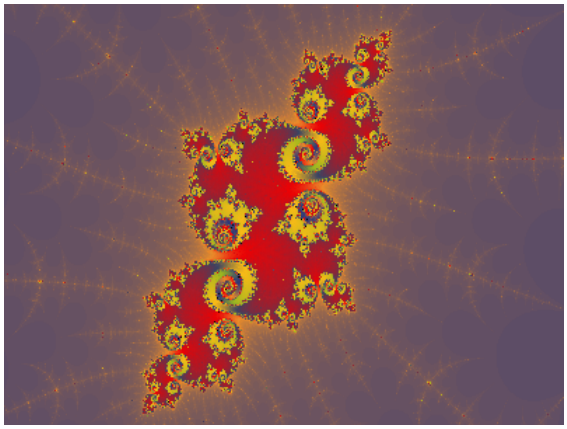
L'image ci contre (type Mandelbrot) est obtenue avec :

- F(z) = sin(z) x z0
- Plan complexe défini par les points bas-gauche (-3.2, -2.0) et haut-droit (3.2, 2.0)
- Module de sortie : 64
- Profondeur : 256



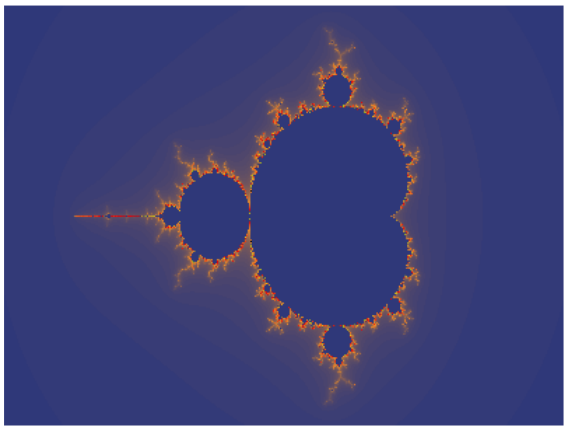
L'image ci contre (type Julia) est obtenue avec :

- $F(z) = z^2 + c$ avec c = -0.750660 + 0.034635 i
- Plan complexe défini par les points bas-gauche (-1.6, -1) et haut-droit (1.6, 1)
- Module de sortie : 4
- Profondeur : 256



L'image ci contre (type Julia) est obtenue avec :

- F(z) = cos(z) x c avec c = 4.72675 + 0.001456 i
- Plan complexe défini par les points bas-gauche (0.1407, 0.1177) et haut-droit (0.1431, 0.1192)
- Module de sortie : 900
- Profondeur : 2500



L'image ci contre (type Mandelbrot) est obtenue avec :

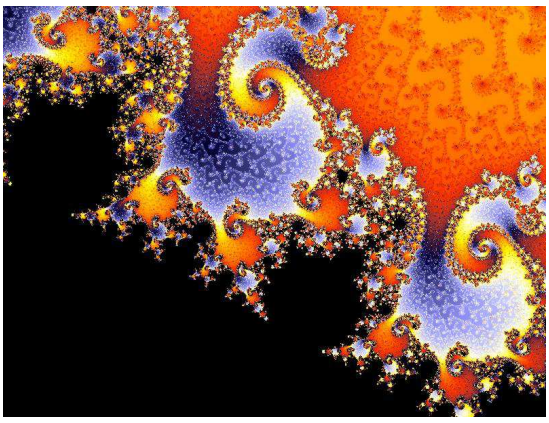
- F(z) = z^2 + z0
- Plan complexe défini par les points bas-gauche (-2.5, -1.25) et haut-droit (1.5, 1.25)
- Module de sortie : 4
- Profondeur : 256

Principes de gestion de la palette de couleurs avec dégradés

L'approche est la suivante :

- On définit un petit nombre P de couleurs (par exemple jaune (#FFFF00), vert (#00FF00), bleu (#0000FF) et rouge (#FF0000)), appelées paliers, et une valeur N correspondant au nombre de couleurs entre deux paliers, de manière à obtenir une transition entre ces deux paliers.
- À partir de ces informations, il faut générer toutes les couleurs correspondantes (au nombre de (P - 1) * N).
- Entre autres, il faut bien faire attention à diviser les paliers selon les 3 composantes R, V et B.
- Une fois toutes les couleurs obtenues, il suffit de colorier un point divergeant avec la couleur ayant pour indice le numéro de l'itération ayant fait diverger la suite (modulo le nombre de couleurs total).
- Directives de codage :
    - Couleurs à stocker sous forme de uint32_t
    - Utilisation des opérateurs de décalage (<< et >>) pour le calcul des composantes R, V et B

Exemple d'image obtenue avec la méthode proposée des paliers de dégradés à partir de couleurs primaires :



Export du fichier image au format PPM

De manière à pouvoir visualiser le résultat, on se propose d'utiliser en sortie le format d'image ppm (Portable Pixel Map). Ce format très simple consiste en un entête spécifiant le type d'image (ex : couleur ou noir et blanc), ses dimensions puis la suite des pixels de l'image ligne par ligne, chaque pixel étant codé sur 3 octets : un pour le rouge (R), un pour le vert (V) et un pour le bleu (B). Un tel format d'images est visualisable par beaucoup de visionneurs, comme par exemple eog (eye of gnome) sous linux/Unix. On utilisera le format de type texte plus simple à gérer et vérifier (http://netpbm.sourceforge.net/doc/ppm.html).

Il est possible ensuite de convertir les fichiers ppm en images jpg en utilisant par exemple la commande linux ppmtojpeg, avec l'option – –smooth=30 pour lisser le résultat.