



ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE

Problème Scientifique Informatique

Projet Traitement d'Images C#

FORESTIER Alexandre, TD G

ESILV- École Supérieure d'Ingénieurs Léonard-de-Vinci

17 Avril 2019

STRUCTURATIONS DES DONNEES

- Edition d'une image :

A la lecture de l'image nous générons tout d'abord un objet import qui va se charger de lire le fichier. Pour en dégager une matrice pixel que l'on récupérera et que l'on insérera dans la classe MyImage.

Nous pouvons à partir de maintenant effectuer toutes les modifications à l'aide de la classe traitement, composée de fonctions static prenant comme principale paramètre l'image en question.

- Génération d'une image :

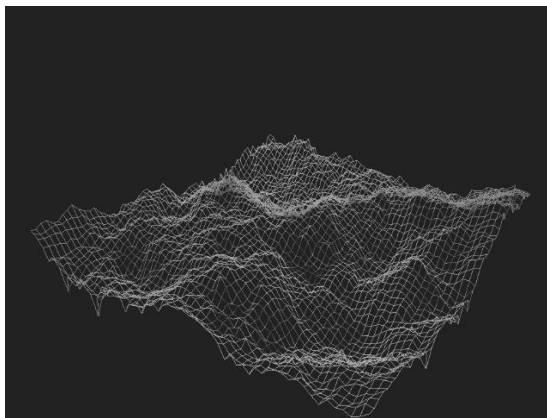
Le seul moyen de générer une image à partir de rien est de créer une fractale, il suffit donc d'appeler une des 2 fonctions static qui composent la classe fractale.

- Exportation de l'image

Pour générer l'image finale il suffira de déclarer un objet export.

MON INNOVATION

A la base je voulais générer ça, théoriquement je n'en suis pas loin. Le problème est que je n'ai pas eu le temps nécessaire pour appliquer tout ce savoir théorique.

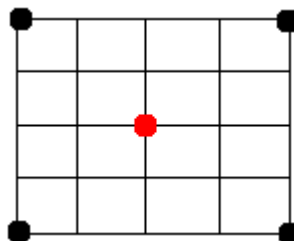
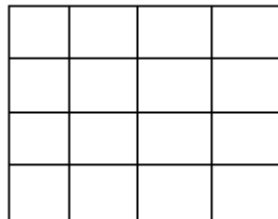


(1)

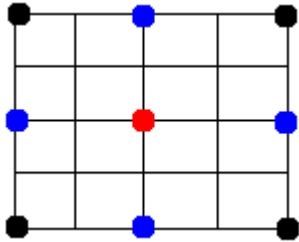


Malheureusement je fais seulement ça (la couleur correspond dans ce schéma à l'altitude du point aux coordonnées i, j , j'utilise l'algorithme de diamant carré :

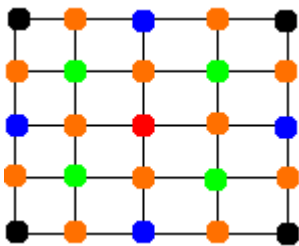
Prenons un exemple simple pour expliquer l'algorithme. Soit une matrice 5x5 :



La phase du diamant consiste à élever le centre de chaque losange formé par le centre et les coins du carré précédent d'une valeur aléatoire (les points en bleu). On peut constater déjà qu'on se retrouve confronté à un problème : il n'y a pas quatre points complet pour former le diamant. Il faut donc tenir compte du cas particulier des bords :



On réitère avec les carrés obtenus(vert), puis les diamants(orange) et ainsi de suite jusqu'à parcourir l'ensemble de la matrice:



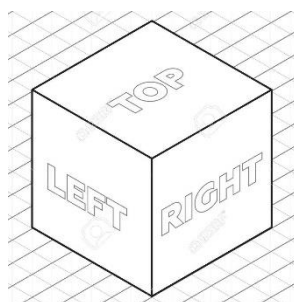
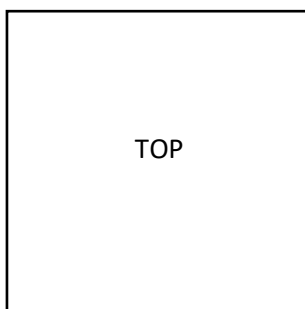
source : <https://hiko-seijuro.developpez.com/articles/diamond-square/>



Cette vue correspond en fait à la vue en 2D des terrains précédents. J'ai passé beaucoup de temps à réfléchir comment faire ceci, finalement j'ai trouver 2 manière de faire une maison et une utilisant wpf (il ne reste plus qu'à les coder

- **Méthode maison** : Elle consiste (visuellement) à insérer $f(i,j)$ dans un repère x,y,z chose que l'on ne fera pas dans le code pour des raisons d'optimisations.

L'objectif étant de passer de



Pour cela nous devons utiliser une matrice de rotation A (de taille 4×4) dont l'expression dépend de α et β on obtient alors :

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = A \begin{pmatrix} x \\ y \\ b(x, y) \\ 1 \end{pmatrix}$$

Puis il suffit de rajouter une matrice de projection et on obtient notre image.

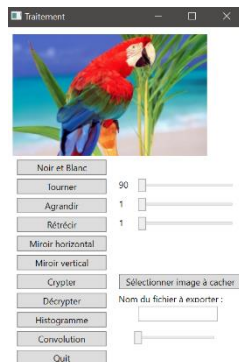
- Méthode WPF : Il suffit de créer un 3D model qui est lui-même composé de meshes (triangles). Le 3D model se compose de au moins 2 collections : une première de n points (x,y,z) et une deuxième de 2n triangles (premier point à relier, deuxième, troisième). On peut rajouter un vecteur normal à la surface ainsi qu'une texture à chaque triangle.

Estimation du temps passé sur le projet : 100h.

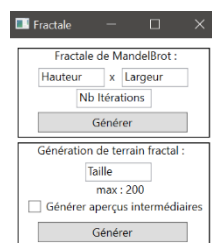
APPLICATION WPF :

La partie wpf se compose de 4 fenêtres :

MainWindow qui correspond à la disjonction présentée au début :



Menu Traitement correspondant au traitement d'une image préexistante



Menu Fractale : permettant de générer les fractales



BIBLIOGRAPHIE :

<https://books.google.fr/books?id=557FevGmvyUC&pg=PA219&lpg=PA219&dq=C%23+show+3D+matrix&source=bl&ots=ZyAoBj58nl&sig=ACfU3U2fKliSbUXOTlee5VLezb5xQvGLjw&hl=fr&sa=X&ved=2ahUKEwif9bLDpc3hAhVCrxoKHZjnAjlQ6AEwC3oEACAcQAQ#v=onepage&q&f=true>

<https://www.codeproject.com/Articles/24727/WPF-3D-Part-1-of-n>