

## La génération procédurale ou comment générer aléatoirement un paysage cohérent

La génération procédurale possède de nombreuses applications dans des domaines variés comme les films, les jeux vidéo, la photographie... J'ai décidé de l'étudier à travers deux algorithmes permettant une génération quasi aléatoire mais toujours contrôlée, afin de créer un paysage homogène : celui des diamants carrés et l'algorithme par transformée de Fourier rapide (FFT) (permettant un meilleur contrôle de la génération). Je donnerai ensuite une application géologique simulant les interactions du paysage avec l'eau ou la chaleur causées par l'érosion.

### Positionnement thématique (phase 2)

*INFORMATIQUE (Informatique pratique), INFORMATIQUE (Informatique Théorique), MATHEMATIQUES (Autres).*

### Mots-clés (phase 2)

Mots-Clés (en français)	Mots-Clés (en anglais)
<i>Génération procédurale</i>	<i>Procedural Generation</i>
<i>Bruit</i>	<i>Noise</i>
<i>Transformée de Fourier</i>	<i>Fourier Transformation</i>
<i>Carte de Hauteur</i>	<i>Heightmap</i>
<i>Algorithme d'érosion</i>	<i>Erosion algorithm</i>

### Bibliographie commentée

La génération procédurale permet de créer de manière aléatoire (ou quasi-aléatoire) des environnements réalistes comme des végétaux (systèmes de Lindenmayer), des textures pour générer des matériaux tels le marbre (Procedural Shading), des générations de terrains complexes (comme dans le jeu vidéo Minecraft) et des paysages sous la forme de cartes 3D. On trouve donc beaucoup de génération procédurale dans les jeux vidéos car elle permet d'obtenir une plus grande variété de décors tout en minimisant l'espace mémoire pris par le jeu. Elle est aussi présente en photographie et au cinéma où, grâce à des techniques avancées, les résultats deviennent aussi réalistes que sur des vraies photographies.

Parmi les algorithmes existant, l'un des plus élémentaires est l'algorithme des Diamants Carrés [1] qui consiste à remplir une matrice carrée à partir de quatre valeurs aléatoires placées aux quatre coins en effectuant des calculs de moyenne et d'ajouts de valeurs aléatoires. La matrice ainsi obtenue est interprétée comme une carte de hauteur (les numéros des cases de la matrice vus comme abscisse et ordonnée («x et y») et la valeur de chaque case comme la valeur hauteur («z»)). Mais, les résultats, quoique convaincants restent incontrôlables. Ainsi, un autre algorithme utilisant la synthèse de Fourier [2] permet d'obtenir des résultats tout aussi réalistes tout en contrôlant le type de terrain [3].

On passe une matrice de nombre aléatoires dans le domaine fréquentiel grâce à la transformée de Fourier rapide. En interprétant les fréquences en fréquences spatiales, on applique un filtre [4]

permettant de garder plus ou moins les hautes fréquences puis on repasse dans le domaine réel et en interprétant la matrice comme une carte de hauteur, on obtient un paysage plus ou moins montagneux.

Afin d'obtenir un rendu encore plus réaliste, un algorithme d'érosion thermique et/ou hydraulique peut être appliqué [5]. En effet, un terrain peut être soumis à plusieurs interactions avec les éléments extérieurs comme l'eau (par la pluie notamment) ou la chaleur (le Soleil) et suivant la composition du terrain, l'érosion est différente. Prendre en compte cet aspect peut donc améliorer le réalisme du paysage final.

L'utilisation de tous ces algorithmes va donc nous permettre de créer un paysage qui, en plus d'être homogène, sera assez réaliste pour être exploité dans différents domaines.

## Problématique retenue

Comment générer de manière aléatoire un paysage homogène et cohérent, plus ou moins escarpé, et en quoi la prise en compte d'un environnement extérieur peut elle améliorer le rendu final ?

## Objectifs du TIPE

- GENERATION: Générer une carte 3D aléatoire représentant un paysage réaliste, homogène , cohérent.
- CONTRÔLE: Trouver un algorithme permettant de mieux contrôler la génération (la rendant ainsi semi-aléatoire) et de créer au choix des reliefs plus ou moins montagneux.
- AMELIORATION: Appliquer un algorithme sur la carte simulant l'érosion thermique et l'érosion hydraulique pour rendre le relief obtenu plus réaliste.

## Abstract

Procedural generation has been widely used in video games or movies and it is still used today. Thanks to two different algorithms, the Diamond-square algorithm and another based on the Fourier synthesis , I managed to create 3D-maps and, even if the creation is random, I can slightly control the kind of land (flat or mountainous) wanted by the user. Finally, in order to increase the realism , I implemented another algorithm that simulates the erosion

## Références bibliographiques (phase 2)

[1] WIKIPEDIA : Algorithme Diamant-Carré :

*[https://fr.wikipedia.org/wiki/Algorithme\\_Diamant-Carre](https://fr.wikipedia.org/wiki/Algorithme_Diamant-Carre), site consulté jusqu'à début décembre 2017*

[2] KEITH LANTZ : Using Fourier synthesis to generate a fractional Brownian motion surface :

*<https://www.keithlantz.net/2011/11/using-fourier-synthesis-to-generate-a-fractional-brownian-motion-surface>, site consulté régulièrement depuis janvier 2018*

[3] DAVID S. EBERT, F. KENTON MUSGRAVE, DARWYN PEACHEY, KEN PERLIN, STEVEN WORLEY : Texturing and Modeling , A Procedural Approach : *Chapitre 16, Morgan Kaufmann, 3 edition (16 Décembre 2002)*

[4] WIKIPEDIA : Pink Noise : *[https://en.wikipedia.org/wiki/Pink\\_noise](https://en.wikipedia.org/wiki/Pink_noise), site consulté régulièrement depuis janvier 2018*

[5] JACOB OLSEN : Realtime Procedural Terrain Generation :  
*<http://web.mit.edu/cesium/Public/terrain.pdf>, site consulté régulièrement depuis décembre 2017*

### Références bibliographiques (phase 3)

[1] XING MEI, PHILIPPE DECAUDIN, BAO-GANG HU : Fast Hydraulic Erosion Simulation and Visualization on GPU : *<https://hal.inria.fr/inria-00402079/document>*

### DOT

[1] *Génération d'une carte utilisant l'algorithme des diamants-carrés : constat qu'aucun contrôle sur la carte n'est possible.*

[2] *Changement de méthode : afin d'avoir un meilleur contrôle, génération d'une carte utilisant un algorithme basé sur la synthèse de Fourier.*

[3] *Décision d'appliquer un algorithme d'érosion aux cartes générées.*

[4] *Implémentation d'un premier algorithme basé sur le travail de Jacob Olsen : résultats absurdes probablement en raison d'erreurs dans le code. Cependant la théorie d'Olsen ne semble pas prendre en compte le temps ou la gravité donc je décide de changer d'implémentation.*

[5] *Réalisation d'un second algorithme basé sur le travail de Xing Mei, Philippe Decaudin et Bao-Gang Hu, permettant un meilleur contrôle et donnant des résultats plus convaincants.*