

# Projet : Génération d'un terrain en 3D

## I/ Introduction

Créer un terrain en 3D, c'est faire le premier pas vers la génération de mondes virtuels, d'où l'importance du concept. De plus en plus, avec l'avancée des technologies, on parvient à « numériser » notre environnement dans des buts aussi divers que les jeux vidéo (bien sûr !) mais aussi la médecine, la construction de bâtiments, les simulations d'expériences scientifiques, le passage de permis de conduire etc.

En savoir plus :

Une opération chirurgicale effectuée à 7000 km de distance :

<https://www.e-sante.fr/premiere-operation-chirurgicale-plus-7-000-km-distance/actualite/657>

Soigner des phobies :

<https://www.essentiel-sante-magazine.fr/sante/traitements-soins/soigner-phobies-realite-virtuelle>

Simulation d'essais nucléaires (depuis plus de 20 ans en France) :

[http://www.irenees.net/bdf\\_fiche-notions-244\\_fr.html](http://www.irenees.net/bdf_fiche-notions-244_fr.html)

Simulation d'une conduite sur route :

<https://www.ecf.asso.fr/Les-plus-ECF/The-Good-Drive>

Tout ceci n'a été possible que grâce à l'amélioration de la puissance des ordinateurs mais aussi par l'élaboration d'algorithmes permettant de retranscrire le monde réel en données binaires d'une manière de plus en plus fidèle.

## II/ Principe de fonctionnement des algorithmes

Les algorithmes permettant de générer un terrain en 3D fonctionnent de la même façon :

- Etape 1 : génération d'une liste de listes de nombres indiquant les altitudes de chaque point grâce à un algorithme.
- Etape 2 : génération d'une image en dégradé de gris à partir des données précédentes. L'altitude minimale sera représentée par un pixel noir, la maximale par un pixel blanc soient des valeurs comprises entre 0 et 255 (code RVB). On peut éventuellement appliquer un traitement à l'image (floutage, netteté etc...) pour améliorer son rendu.
- Etape 3 : traduction des valeurs des pixels pour générer le terrain.

Exemple de génération d'un terrain avec une montagne au centre :

### Etape 1

Création et remplissage d'une variable de type `tableau[i][j]` où les indices `i` et `j` représentent les abscisses et ordonnées du terrain.

### Etape 2

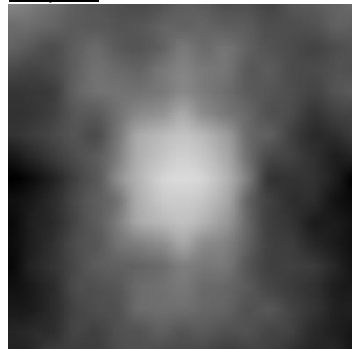
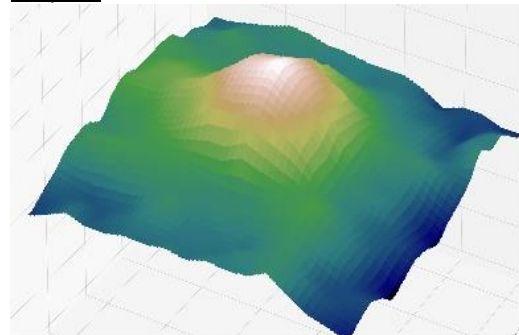


Image en dégradé de gris

### Etape 3



Génération finale du terrain

# III/ Présentation de l'algorithme de Perlin

## 1/ L'algorithme de Perlin ou le début des images de synthèse

Ken Perlin, dans le cadre de la réalisation du film *Tron* (1982) comportant des scènes en images de synthèse, se retrouva confronté aux limitations mémoire des machines de l'époque, ne permettant pas de stocker de nombreuses et imposantes textures.

Il chercha alors à générer ces textures par le calcul, que l'on nomme *textures procédurales*. Elles offrent l'avantage d'occuper peu de place en mémoire et aussi une certaine flexibilité.

## 2/ Philosophie de l'algorithme de Perlin

L'idée générale est de s'appuyer sur les propriétés du son, notamment la différence entre un son dit « pur » (un « la » au diapason ou à la sonnerie téléphonique) et un son dit « complexe » qui représente le timbre unique de chaque type d'instrument.

Un son « complexe » à l'oreille se révèle beaucoup plus agréable qu'un son « pur ». Le principe est le même avec l'algorithme de Perlin qui va permettre de proposer des terrains plus esthétiques.

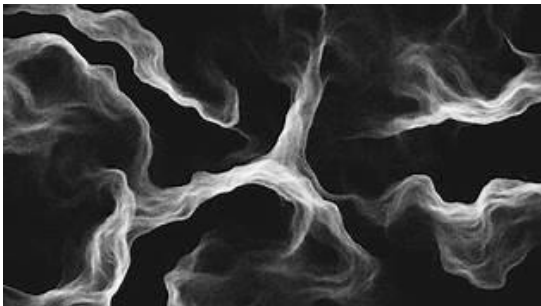
Plus d'informations ici (première partie) : <https://tpeinfluancemusique.webnode.fr/le-son2/les-sons-purs/>

## 3/ Utilisation actuelle de l'algorithme

Il est très utilisé pour l'élaboration d'éléments de mondes virtuels comme par exemple les terrains, les nuages, la houle etc.

On le retrouve également dans la génération de textures à 2D mais aussi en 3D qui permettent des animations.

Quelques exemples :



Effets type « brume ou fumées »



Texture type « bois »

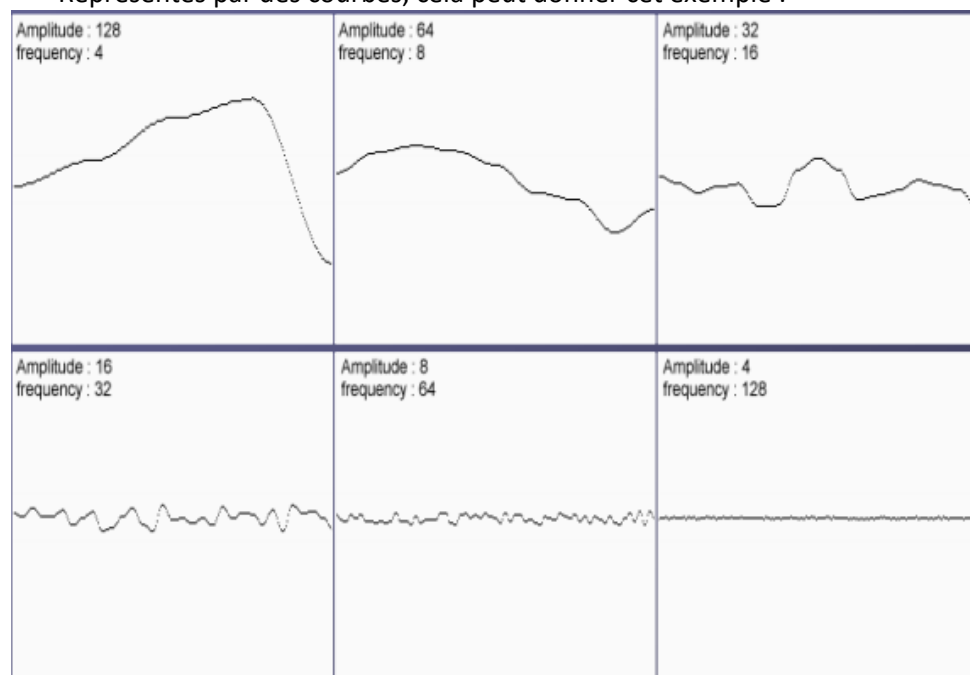


Un continent

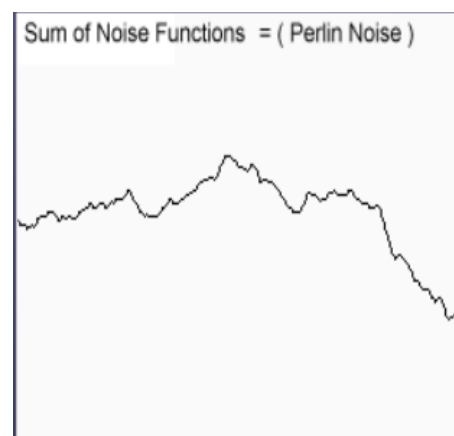
## 4/ Fonctionnement de l'algorithme de Perlin

Cet algorithme consiste à additionner plusieurs tableaux d'éléments aléatoires avec des poids différents.

Représentés par des courbes, cela peut donner cet exemple :



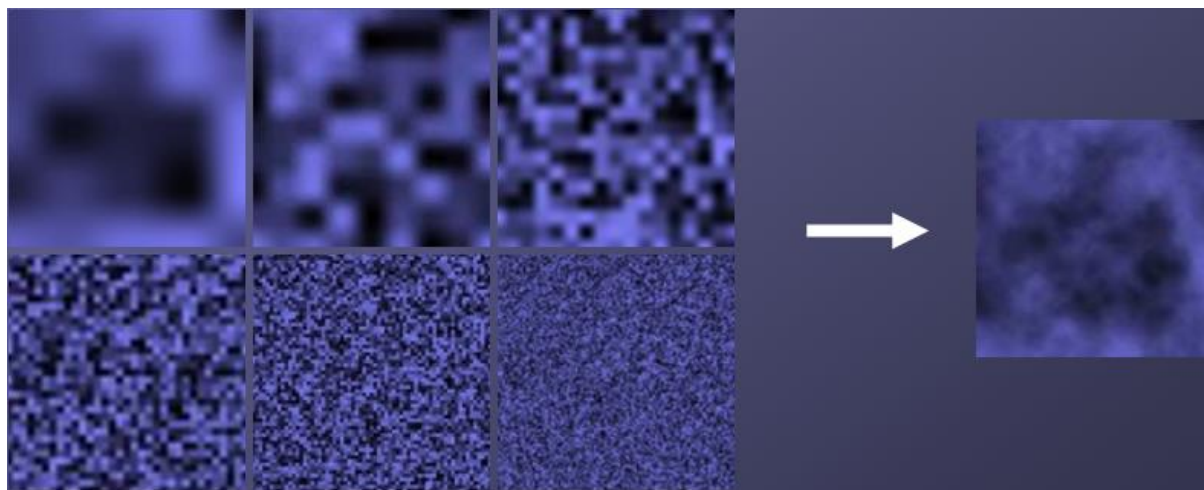
En additionnant chaque courbe, on obtient ceci :



La **fréquence** est le nombre de points générés aléatoirement, tous les autres sont déterminés en **interpolant les valeurs des points les plus proches** (principe de l'algorithme des k plus proches voisins ).

On remarque que l'allure générale de la première courbe est conservée : c'est normal, c'est celle qui compte le plus et c'est là-dessus que l'on peut jouer pour déterminer les résultats souhaités.

Sous forme d'images, on obtient, par le même principe :



Il ne reste plus qu'à traduire la teinte de gris de chaque pixel en altitude : plus il est clair, plus elle est élevée.