

Le sonar de l'infini

Mysaa (Samy AVRILLON)

18 mai 2021

Table des matières

I	Problématique	2
II	Génération	2
1	Bruit de Perlin	2
2	Bruit fractal	3
III	Définition des paramètres et de presets	3
IV	Ouverture	3

Première partie

Problématique

L'objectif de ce TIPE est de générer des fonds marins aléatoires, réalistes ou au moins 'utiles'. Les applications possibles seraient, par exemple

- La génération de photographies de synthèse de paysages aléatoires, dans le cinéma par exemple. Les paramètres seront ainsi déterminés de manière à créer les paysages les plus réalistes possible, ou modifiés pour servir un message artistique (par exemple, demander un sol plus imparfait).
- De la génération procédurale dans des jeux vidéos. L'exemple le plus pertinent est le jeu vidéo *Minecraft*, où le joueur évolue dans un monde généré aléatoirement grâce à une graine (*seed* en anglais) de manière déterministe (deux mondes générés avec une même graine seront rigoureusement identiques). De plus, le moteur de génération est géré de manière à ce que l'on puisse lui demander de générer une zone 'loin' sans qu'il n'ait à générer toutes les zones à partir de l'origine. Enfin, la carte générée est potentiellement infinie (limité bien sûr par la taille de stockage allouée aux entiers et aux flottants).
- De la génération de fonds pour des tests en simulation d'IA de robot sous-marin (voir projet de Quentin SOUVIGNET) ou pour tester des simulations de tsunamis.

De ces possibles applications, je tire un « Cahier des charges » de mon générateur :

- Laisser accès au plus de paramètres possible, pour une meilleure personnalisation.
- Permettre une génération procédurale, déterministe et non liée à l'origine.
- Limiter la complexité en temps du programme, éventuellement la complexité spatiale.

Deuxième partie

Génération

Je vais présenter dans cette partie les différents algorithmes implémentés, leurs applications possibles, leurs caractéristiques.

1 Bruit de Perlin

Il s'agit d'un bruit très flou (voir Figure 1). Il peut être utilisé pour la génération d'une carte grande échelle, indiquant les variations globales du terrain. Je vais maintenant donner une description simplifiée

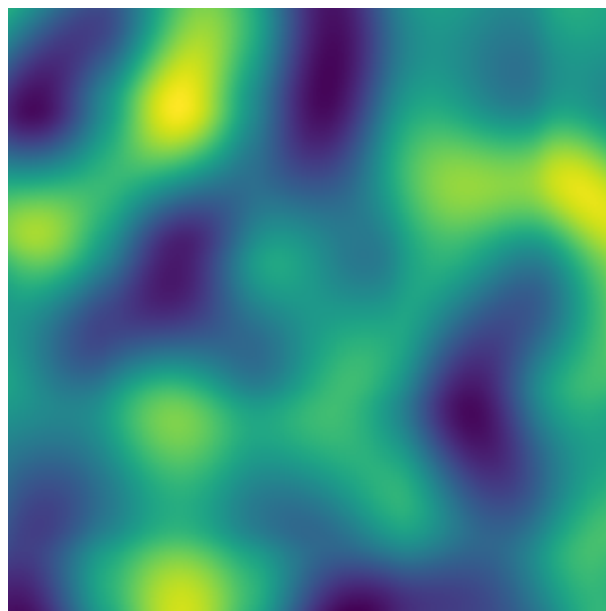


FIGURE 1 – Exemple de génération d'un bruit de Perlin sur une zone de 5x5

de fonctionnement de l'algorithme : Il génère d'abord une carte de gradients 2D de norme unitaire de direction et sens aléatoires (vecteurs du cercle trigonométrique), positionnés en chaque noeud entier. Ensuite, il « lance une nappe » qui va prendre une forme imposée par les gradients. L'algorithme met en jeu une fonction d'interpolation pour déterminer les valeurs hors des noeuds, ce qui peut être un paramètre sur lequel agir.

L'avantages de cet algorithme est sa rapidité et sa faible complexité spatiale.

2 Bruit fractal

Cet algorithme génère des droites de manière aléatoires, ce qui sépare pour chaque droite le plan en deux, et surélève l'un des deux cotés. Ce bruit est beaucoup plus « granuleux » que le brut de perlin, et semble par conséquent plus réaliste. Le problème est que cette méthode est incompatible avec l'infinité de la map. On ne peut en effet pas charger l'infinité des droites du plan pour générer un carré (une célèbre équation annonce que $\forall x \in \mathbb{R}, x < +\infty$). J'ai donc modifié l'algorithme en restreignant l'action des droites progressivement (grâce à une fonction d'interpolation, paramétrable elle aussi) Voir Figure 2 pour deux exemples avec plus ou moins de droites par region. L'inconvénient de cette methode est qu'elle devient vite coûteuse lorsque l'on augmente le nombre de droites.

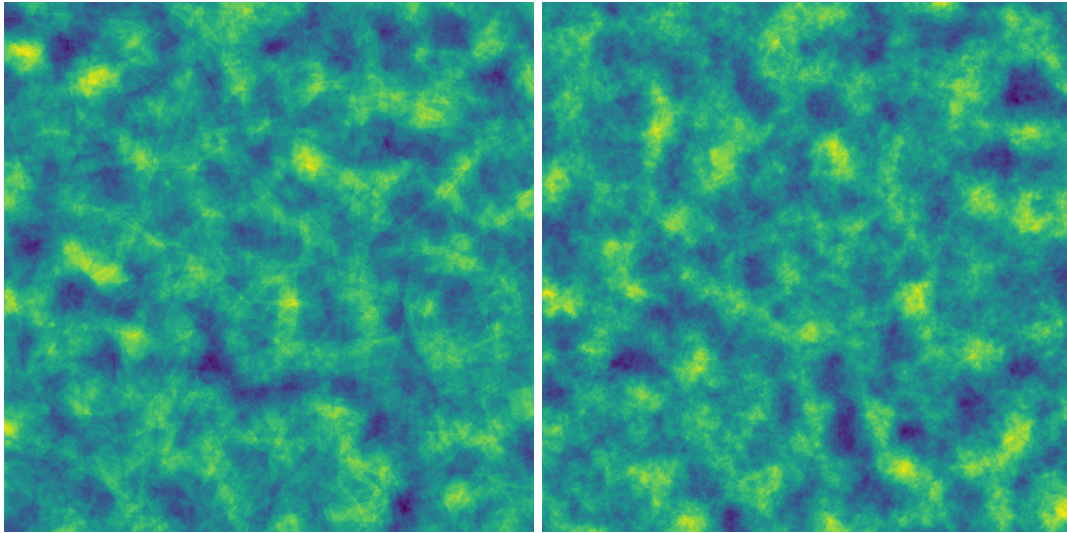


FIGURE 2 – Bruit fractal avec peu à gauche et beaucoup à droite de droites par chunk

Troisième partie

Définition des paramètres et de presets

A faire ...

Quatrième partie

Ouverture

Je vais ici lister différents extensions pouvant être apportées à l'étude, si le sujet devient complètement traité.

- Créer un moteur de générations d'images des fonds marins générés précédemment, notamment avec la technique du ray-tracing.
- Textures de l'eau aléatoires, des coraux aléatoires, avec une texture ayant l'air organique ...
- Créer un moteur physique simplifié permettant à un joueur virtuel de se déplacer dans l'océan virtuel. Déterminer le meilleur rapport coût algorithmique/réalisme