

Génération procédurale de ciel

Introduction

À l'heure où les jeux sont de plus en plus vastes et gourmands en octets, la génération procédurale, quelque peu abandonnée lorsque les capacités de stockages ont augmenté, a refait surface. Il semble important pour bien comprendre les problématiques et le sujet de commencer par définir ce qu'est la génération procédurale, pour ensuite s'attarder sur le sujet qui nous est propre, la génération procédurale de ciel. On parlera ici de la génération de textures, notamment pour la skybox, de la lumière et de sa gestion, et enfin des potentielles animations du ciel.

Mot clés : **procédural, génération, jeux vidéo, environnement, ciel**

Qu'est ce que la génération procédurale ?

Une donnée générée procéduralement, en opposition avec une donnée générée manuellement, est une donnée générée par un algorithme. Il est possible d'introduire du pseudo random lors de la génération, cela a plusieurs avantages. Le premier



avantage est la taille des données, en effet, sachant qu'en fixant un seed, avec les mêmes paramètres d'entrée, on obtient la même sortie, il est possible de ne rien stocker, et de tout générer lorsque l'on en a besoin. On pourra prendre l'exemple des textures, qui sont de plus en plus volumineuses, notamment avec l'arrivée de la 4k. La génération procédurale de textures permet de sauver un très grand espace disque. On pourra citer la

demoscene, qui avec des fichiers de seulement quelques kilo-octets, où la totalité des données est générée de façon procédurale, le terrain, les textures, le son...

Un autre avantage évident est la diversité du contenu, avec du random, le nombre de déclinaisons est potentiellement infini. Cela permet d'une part d'avoir des mondes plus



vastes, en déchargeant les designers, mais également de rendre le monde moins prévisible.

Le désavantage principal est la quantité de calcul nécessaire pour générer la donnée. C'est d'ailleurs souvent un équilibre à trouver, et notamment dans les jeux vidéos où dans les performances sont critiques.

Quels avantages pour le ciel ?

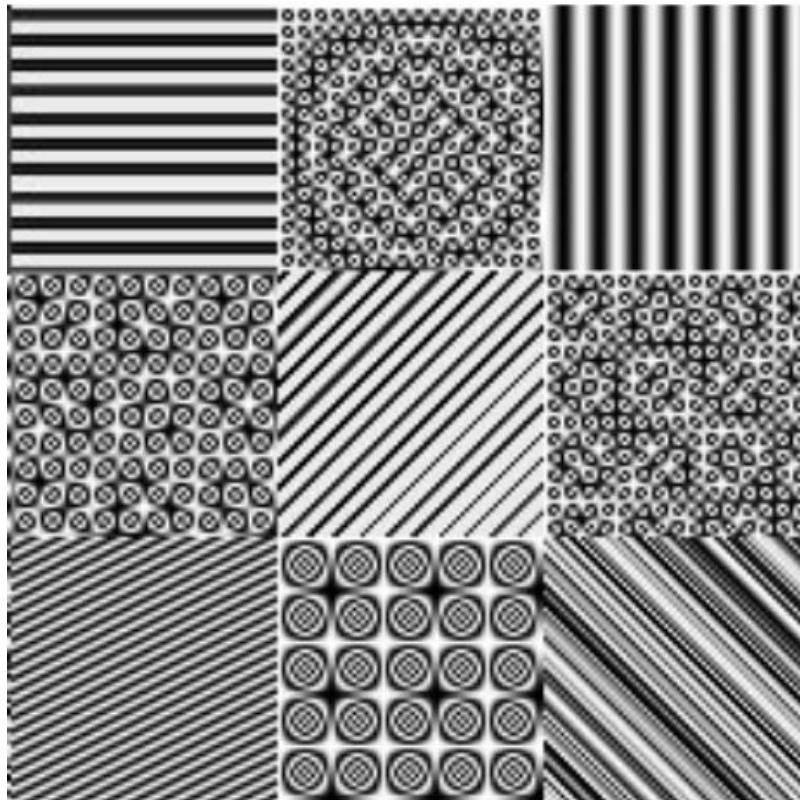
Il semble difficile de parler de ciel sans parler d'ambiance, c'est en effet souvent du ciel que vient la lumière en extérieur, et c'est en grande partie de la lumière que provient l'ambiance. La génération procédurale permet d'adapter finement l'ambiance de la scène aux événements du jeu. C'est un bon moyen d'augmenter l'immersion du joueur.

Génération des Textures

L'affichage du ciel et la gestion de la lumière ambiante dans une scène est aujourd'hui pris en charge par les skybox. Une skybox est un cube ayant, sur ses faces intérieur, un décors projeté. Elle est centrée par rapport à la caméra (et se déplace en même temps qu'elle) ce qui donne l'effet que le décors est infini.

Il est possible d'appliquer différentes textures aux faces de la skybox et de changer la couleur ou la position de la lumière pour faire varier l'ambiance. Le ciel n'étant en général pas un élément de décors qui change brutalement d'apparence, la nouvelle texture générée doit être proche de la texture précédente, on pourra citer un coucher de soleil, où la position du soleil sa couleur, et donc la couleur de la lumière, varie progressivement. Mais avoir autant de textures poser problème car lorsque l'on a des ambiances proches, il semble absurde de créer autant de textures qu'il y a de variations d'ambiance. Une solution pour régler ce problème est d'utiliser un algorithme génétique pour générer la nouvelle ambiance en fonction des ambiances déjà utilisés.

L'algorithme génétique va, à partir des éléments qui composent les textures précédentes, réaliser des croisements de ces différents éléments pour générer une nouvelle texture proche des autres textures.



Génération de texture avec un algorithme génétique

L'image ci-dessus un exemple de textures générée avec un algorithme génétique, l'image du haut au milieu est un mélange de l'image en haut à gauche, en haut à droite et l'image de gauche (avec différents facteur de corrélation).

Dans certaines situations, on peut utiliser l'aléatoire pour générer le ciel en plaçant certains éléments de manière aléatoire sur un fond uni (par exemple un ciel étoilé).

Gestion de la lumière

Il existe deux grandes manières de gérer la lumière. La première est d'allumer des lumières et de calculer leur effet sur l'environnement à chaque frame, appelée



lumière dynamique, c'est une manière qui est très coûteuse, mais qui permet d'avoir une lumière en mouvement. L'autre manière beaucoup plus économique, puisqu'elle ne demande pas de calcul supplémentaire à chaque frame, est d'avoir une lightmap, c'est à dire la

réflexion de la lumière pré-calculée et stockée dans une texture. La lumière calculée à chaque frame est de toute façon procédurale, nous allons donc nous intéresser à la possibilité d'une génération procédurale de lightmap. Cela permettrait d'augmenter grandement les performances pour le rendu de la lumière ambiante dans les terrains générés eux aussi de façon procédurale.



On peut déjà, lors de la génération procédurale d'objets du décor par exemple, les générer avec une lightmap, si on connaît à l'avance la position des sources de lumière et qu'il ne s'agit pas d'une lumière ponctuelle, le problème ne se pose donc que lorsque la lumière est amenée à varier. Mais même cette méthode n'est pas parfaite, car il n'est pas possible d'intégrer les ombres projetées par les autres objets dans la lightmap. Il n'y a en vérité pas de bonne méthode de génération procédurale de lightmap, car le principe même de génération procédurale est contraire au pré-calcul.

Animations

Une animation procédurale est une animation dont les déplacements sont calculés en temps réel en suivant un certain nombre de règles.

Différentes animations peuvent être affichées dans un ciel : le déplacement des nuages, des étoiles filantes dans un ciel étoilé ou des éléments de météo (pluie, neige, grêle...).

Il est possible de faire apparaître (de manière temporaire ou pas) des objets dans le ciel se déplaçant selon une certaine fonction (en ligne, en cercle, de manière aléatoire).

On peut également agir sur la fréquence d'apparition des éléments.

Pour l'affichage des nuages, il est possible de les dessiner sur les textures de la skybox et de les modifier régulièrement en combinant différents "images" des nuages.

Pour générer ces textures de nuage, on crée des images de “bruit” de manière aléatoire (variation des niveaux de gris) en taille 32x32 que l’on smooth et que l’on agrandi à 256x256, l’image obtenue est appelée octave. On crée plusieurs images en faisant varier la taille de l’image et le nombre d’image qui compose l’octave. Il faut ensuite superposer les différents octaves créés entre eux pour augmenter les détails des nuages.

La texture est ensuite appliquée au ciel.

Pour donner l’impression de déplacement, il faut déplacer (à des vitesses différentes) certains octaves de la texture en modifiant leur position u et v.

On peut également simuler le développement des nuages dans le ciel en ajoutant de nouveaux octaves à l’octave actuelle toutes les x secondes.



Sitographie

<http://gamedev.stackexchange.com/questions/9993/what-does-it-mean-to-bake-lights>

<http://www.markmark.net/dissertation/harrisDissertation.pdf>

<http://maverick.inria.fr/Publications/2010/HN10/RAPPORT.pdf>

http://acegikmo.com/shaderforge/wiki/index.php?title=Procedural_Skybox

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4150415>

http://www.flipcode.com/archives/Light_Mapping_Theory_and_Implementation.shtml

<http://www.shy.am/wp-content/uploads/2009/01/realistic-real-time-skies-shyam-guthikonda.pdf>