I- Notre scène : La mégastructure

Le but initial de notre projet était d'implémenter un algorithme appelé wave function collapse pour générer des structures procédurales.

Notre scène finale est en grande partie inspiré des œuvres de MC Escher et Tsutomu Nihei, elle consiste en plusieurs salles imbriquées traversés par des escaliers et des ponts.

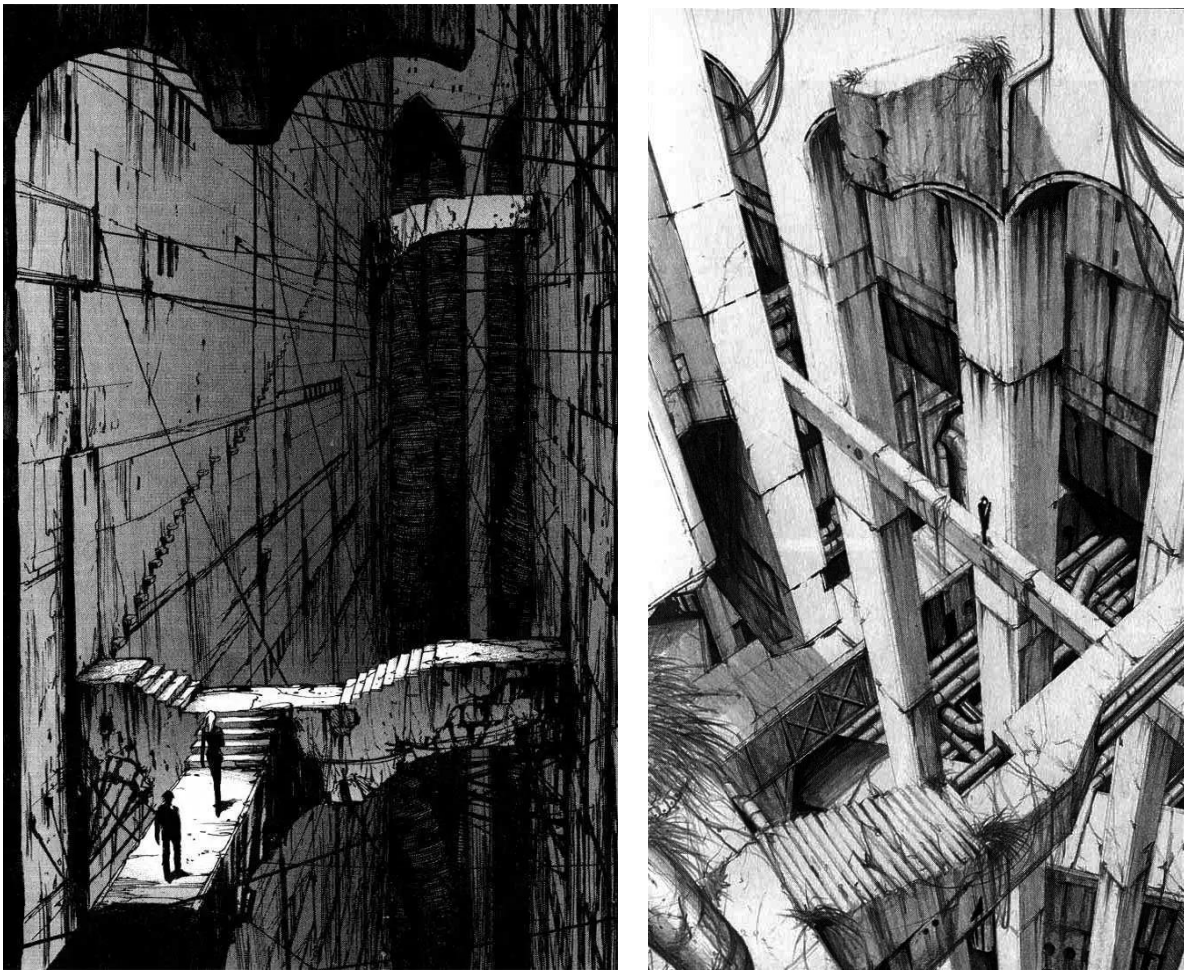

Figure 1 Images de référence, mégastructures du manga BLAME de Tsutomu Nihei

II- L'algorithme Wave Function Collapse

L'algorithme permet de créer un pavage sur une grille à l'aide d'éléments qu'on appellera modules qui possèdent des règles de compatibilités entre eux.

Dans notre cas, nous avons :

- Une grille en deux dimensions (matrice «coefficients» n*n où chaque élément est un tableau contenant les nom des modules possibles à cet emplacement)
- m modules différents associés à des poids qui représentent leur probabilité d'apparaître
- Une matrice « compatibilities » de taille m*m où l'élément[i][j] est un tableau contenant les directions de compatibilité entre le modules i et j (par exemple si compatibilities[0][1] == {« up », « left »} alors le module 1 peut se mettre en haut ou à gauche du module 0.

Initialement, chaque case peut contenir les 8 modules. L'algorithme se décompose en deux phases, la première itère sur l'ensemble des cases de la grille pour trouver la case réalisant le minimum de l'entropie de Shannon (c'est à dire la case où le choix d'un module permis ceux restant a le moins de conséquences) on tire ensuite un des modules disponibles à l'aide de leur poids. Une fois cette case déterminée, on applique la fonction « propagate », qui propage récursivement les conséquences du choix du précédent module aux cases adjacentes.

L'algorithme se termine lorsque toutes les cases sont déterminées, c'est à dire quand le tableau « coefficients » ne contient que des tableaux de taille 1.

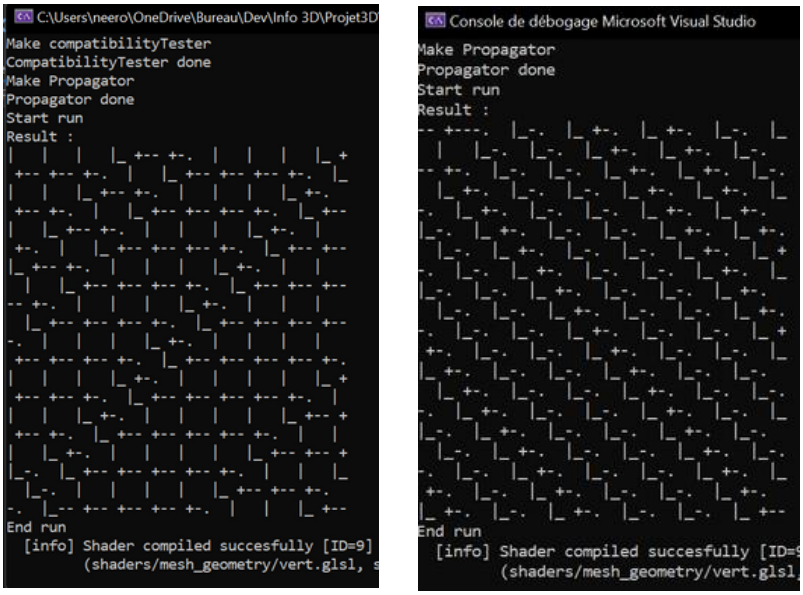

Figure 2 exemples de grilles de module obtenus

On place ensuite des blocs modélisés sur blender qui correspondent à chaque module sur la grille (plusieurs blocs peuvent correspondre au même module dans ce cas on tire le bloc aléatoirement).

III- Brouillard

Pour renforcer le sentiment d'être perdu dans une immense structure, nous avons mis en place du brouillard grâce au shaders, on ajoute une valeur de blanc à la couleur finale en fonction de la distance

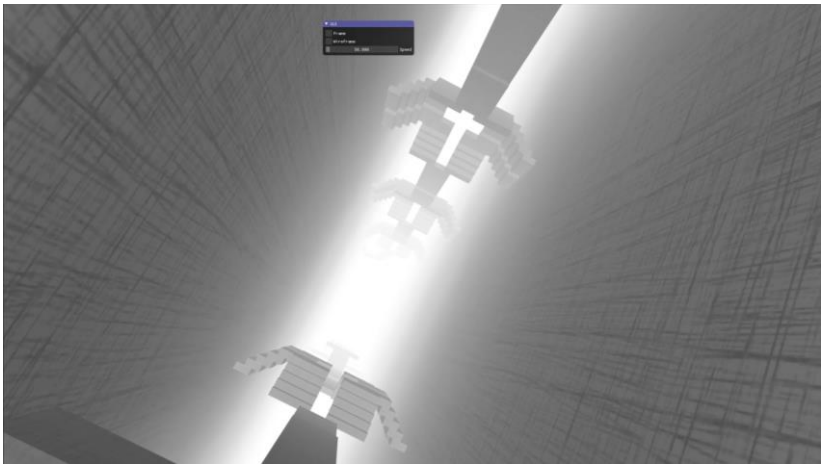

Figure 3 Effet de brouillard

IV- Animation dynamique

Au centre de la structure se situe un fluide visqueux en lévitation qui bat comme un cœur, ce résultat a été atteint avec des blobs. 5 masses reliées par des ressorts donnent la position des centres des fonctions blobs et une surface implicite est tracée à chaque image après l'application de l'algorithme marching cubes.

Cette manière de faire est très coûteuse et peu adaptée au temps réel mais le résultat est fluide en limitant la résolution de la grille utilisée lors de marching cubes.

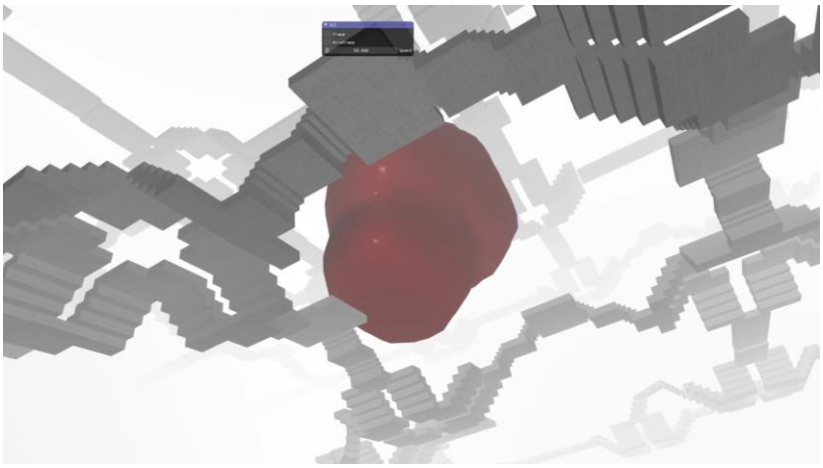

Figure 4 blobs évoluant dynamiquement

V- Améliorations

La caméra peut être améliorée, mais son implémentation actuelle permet un sentiment de flottement et de constater plus facilement la symétrie de la scène.

De plus certains modèles 3D faits sur blender possèdent de mauvais uv et on certaines normales inversées.

Pour aller plus loin, on peut étendre l'algorithme à une grille en 3 dimensions et augmenter le nombre de modules (voir https://marian42.de/article/wfc/ pour 100 modules).