Création Blocs

Pour la création des blocs, nous avions comme contrainte qu’un blocs soit composé de plusieurs autre sous-bloc (bloc élémentaire). Dans un premier temps, nous avons donc générer des cubes composer de plusieurs sous-cube.

Par la suite, nous voulions que nos blocs possèdent des formes aléatoires. Pour cela, nous n’avons plus qu’a enlever aléatoirement des parties du cube pour avoir des formes aléatoire, pour mettre en place ce fonctionnement on utilisera la méthode random() afin de déterminer si un sous-bloc doit être placer ou non.

Gestion Collision

Pour la gestion des collisions, Jmonkey utilise le moteur jbullet. Ce moteur implémente déjà tout un système de physique avec gravité et autres contraintes.

Dans notre cas, nous voulions que les blocs est un comportement dynamique (le bloc subit la gravité et les collisions). Il fallait donc appliquer la physique à notre cube. Cependant, nous avons fait face à un problème au niveau du moteur physique. Effectivement, si on applique la gravité sur le nœud celui-ci génère un volume englobant qui correspond au volume du cube entier. De plus, le centre de gravité de notre cube n’est pas forcement bien placé lors de la mise en place de la physique sur le nœud. Au final, le comportement de nos blocs n’était pas réaliste : par exemple un cube en forme de triangle ne tomber pas sur le coté.

Nous avions pour solution d’appliquer la physique aux sous-blocs, puis dans un second temps de les liée par une contrainte. Ce système fonctionne, mais cela induit un grand nombre d’opération (vérification des contraintes de liaison + collisions entres les sous-blocs + gestion gravité sur tous les objets) ce qui à pour effet de ralentir de façon importante le fonctionnement de l’application.