Rapport de projet de ISIR

BLANCHARD Allan, COUTY Antoine

Avril 2022

1 Structure du projet

Notre archive est composée de ce rapport et de 7 fichiers ".pde" qui contiennent :

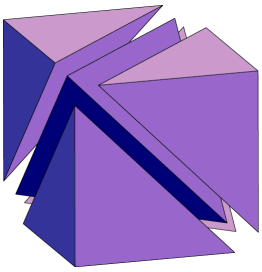
- "Metaballs.pde" : nos variables de configuration du projet, des variables globales (liste des primitives, pavage, ...) et les fonctions setup et draw.
- "primitive.pde" : la classe abstraite Primitive dont doit hériter tous les objets de la scène.
- "primitive_metaball.pde" : la classe Metaball héritant de Primitive.
- "primitive_metaplane.pde" : la classe Metaplane héritant de Primitive.
- "point.pde" : la classe Point qui représente les différents sommets des tétrahèdres pavant l'espace.
- "tetrahedra.pde" : la classe Tetrahedra qui permet de paver notre espace.
- "utils.pde" : toutes nos fonctions utilitaires.

2 Partage des tâches

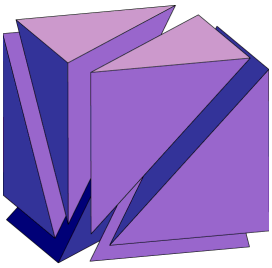
Comme nous avons repris le code d'Allan pour les classes : Primitive, Metaball et celui Metaplane fait durant les séances de cours, il ne nous restait plus qu'à paver l'espace à l'aide de tétrahèdre et de trouver puis afficher les intersections entre les objets de la scène et le pavage de l'espace. Allan s'est chargé de construire la scène et de représenter, stocker et paver l'espace. Antoine s'est chargé de trouver et afficher les intersections entre le pavage et les objets de la scène.

3 Pavage de l'espace

En cours nous avons vu qu'un cube pouvait être découpé en 5 ou 6 tétrahèdres.



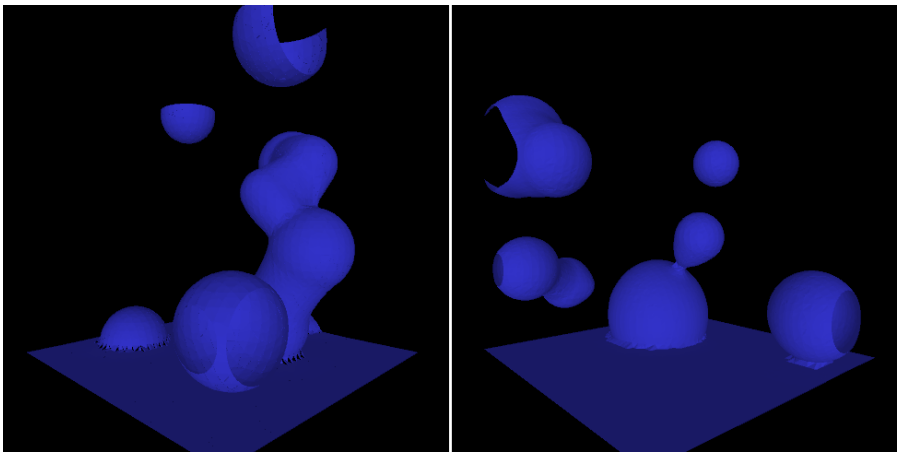
cube coupé en 5 tétrahèdres[1]



cube coupé en 6 tétrahèdres[1]

De ce fait pour pouvoir paver notre espace avec des tétrahèdres il nous faut d'abord paver notre espace avec des cubes. Nous commençons donc par définir notre espace comme étant une suite de points tous espacés par la même distance sur les trois axes[2] que nous stockons dans le tableau 3D "points".

Ensuite en prenant les points huit par huit (que nous interprétons comme étant les sommets d'un cube) nous découpons ce sous-espace en tétrahèdres. Nous avons implémenter puis tester les deux versions et préférons finalament découper les cubes en 6 tétrahèdres plutôt qu'en 5 car il y a certes davantage de calculs à effectuer mais il y a beaucoup moins d'artefact que nous soupçonnons être dû au triangle central.



Résultat avec les cubes coupés en 5 tétrahèdres à gauche et 6 à droite[3]

[1]Ces images sont tirées de ce site.

[2]Distance réglable avec SIZE_VERTICE et nombre de points par axes réglable avec NB_VERTICES.

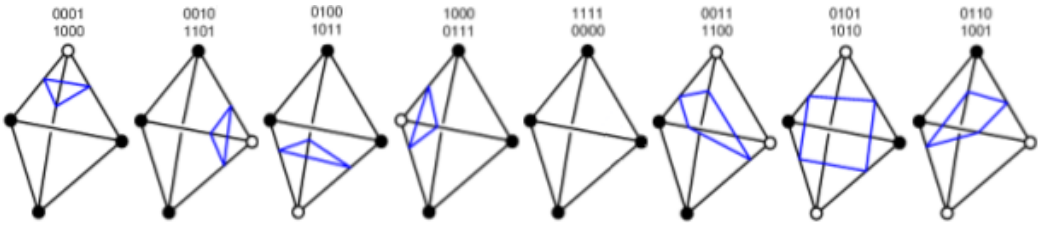
[3]Artefacts et bug au niveau des triangles sortant du sol à gauche et pas à droite.

Puis pour chaque itération de draw il ne nous reste plus qu'à vérifier pour chaque point s'il est contenu ou non par un objet de notre scène, et mettre à jour l'affichage des tétrahèdres en conséquence.

4 Intersection des objets et du pavage

Dans la classe tétrahèdre nous stockons des pointeurs vers ses 4 sommets, ce qui nous permet de savoir à tout moment si un sommet est contenu ou non dans un objet. De ce fait pour trouver les intersections entre nos objets et notre tétrahèdre nous commençons par transcrire "l'état" dans lequel est notre tétrahèdre en fonction de si ses sommets sont effectivement contenus dans un objet ou non. Nous notons 1 si c'est le cas, sinon 0. Nous ajoutons ensuite les bit de chaque sommet à l'aide d'un "ou" logique et nous obtenons un chiffre de 4 bits.

16 états sont alors possibles (de 0000 à 1111). Comme vu en cours nous pouvons nous ramener à 8 cas distincts puisqu'il y a une sorte d'effet miroir dans tous ces cas[4].



Etats possible des tétrahèdres

Nous voyons bien avec la figure ci-dessus que nous avons trois possibilités d'affichage selon les configurations : soit un triangle, soit un quad, soit rien. Mais pour afficher un triangle ou un quad il faut encore trouver une approximation des points d'intersection entre notre tétrahèdre et l'objet. La technique que nous utilisons est la même que celle utilisée dans la démo du marching square. Simplement nous utilisons une version itérative plutôt que récursive.

[4]Par exemple le cas 0 (0000) et 15 (1111) sont deux des cas similaires. Dans les deux cas il n'y a pas d'intersection avec les objets de la scène puisque tous les points du tétrahèdre sont tous contenus ou non dans un objet.