

Moteur de jeux

Beldjilali Maxime – Caro Thomas

Sommaire

Introduction

Entité

La physique

- ReactPhysics 3D
 - Dynamique
 - Collision
 - Broad Phase
 - Narrow Phase
 - Résolution

La destruction de maillage en 2 dimensions

- Destruction
- Diagramme Voronoi
- Algorithme de Fortune
- Génération de points
- Création d'entités

Piste d'amélioration :

- Maillage en 3 dimensions
- Meilleur rendu

Introduction

Fonctionnalités de base :

- Caméra
- Chargement maillage
- Création maillage
- Terrain par Heightmap
- Rendu Phong

Entité

- Maillage
- Transform(Personnelle et ReactPhysics3D)
- Mouvement(Personnelle)
- Rigidbody(ReactPhysics3D)
- Collider(ReactPhysics3D)

Physique

Dynamique

- ReactPhysics3D
- RigidBody
- Dynamique Newtonienne
- Gravité
- Application de force

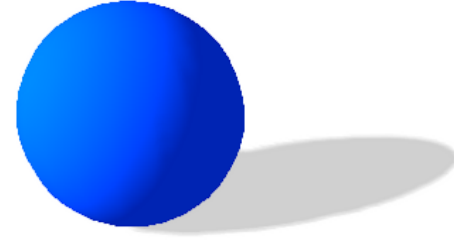
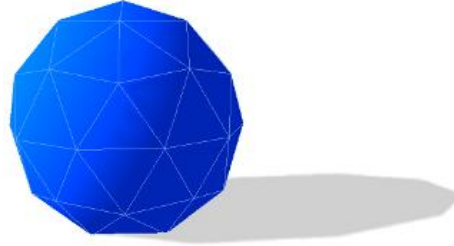
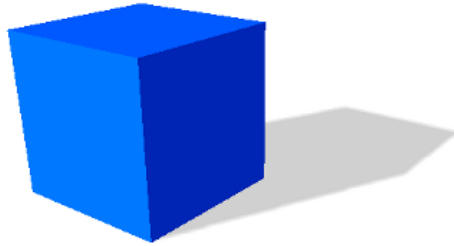
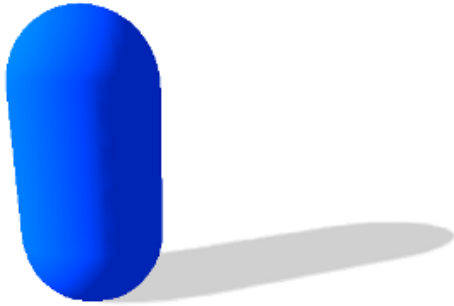
$$p_o(t + \Delta t) = p_o(t) + v\Delta t$$

$$v(t + \Delta t) = v(t) + a\Delta t$$

$$F_{net} = m * a$$

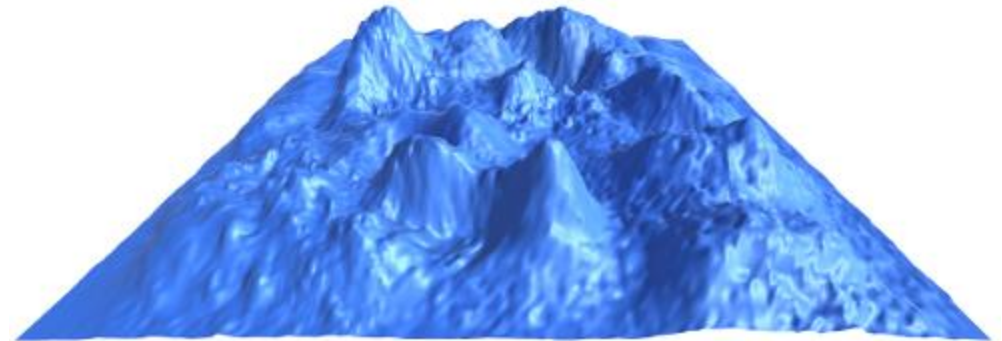
m est le poids de l'objet et a l'accélération

Physique



Collision

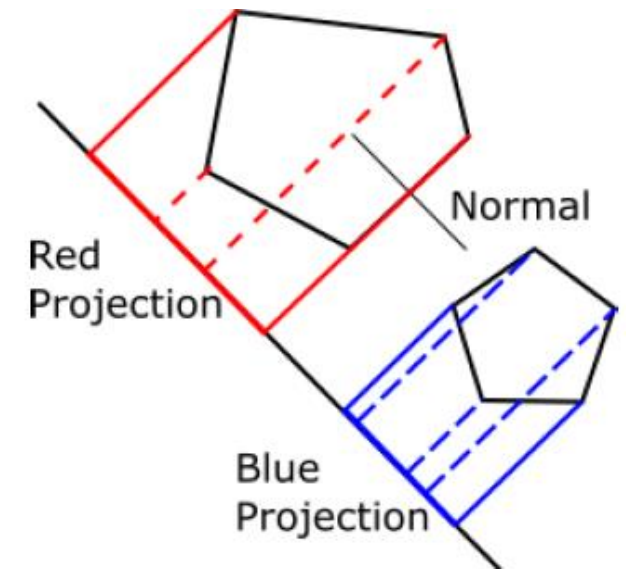
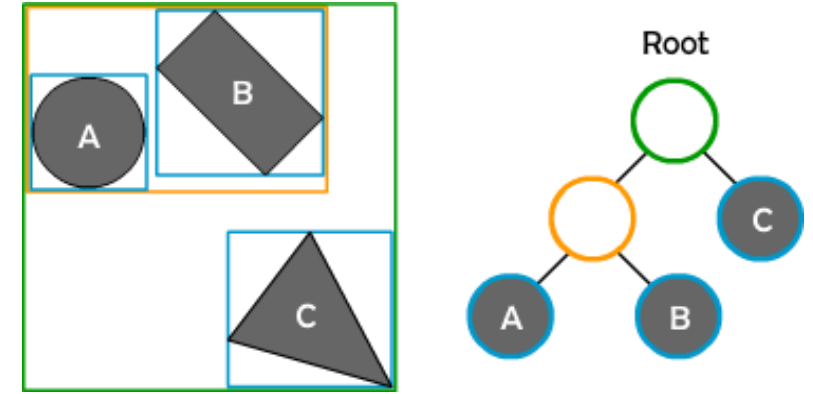
- Création d'une forme
- Collider à partir de forme
- Type de matériaux
- Association au Rigidbody



Résolution de collision

Solveur ReactPhysics3D :

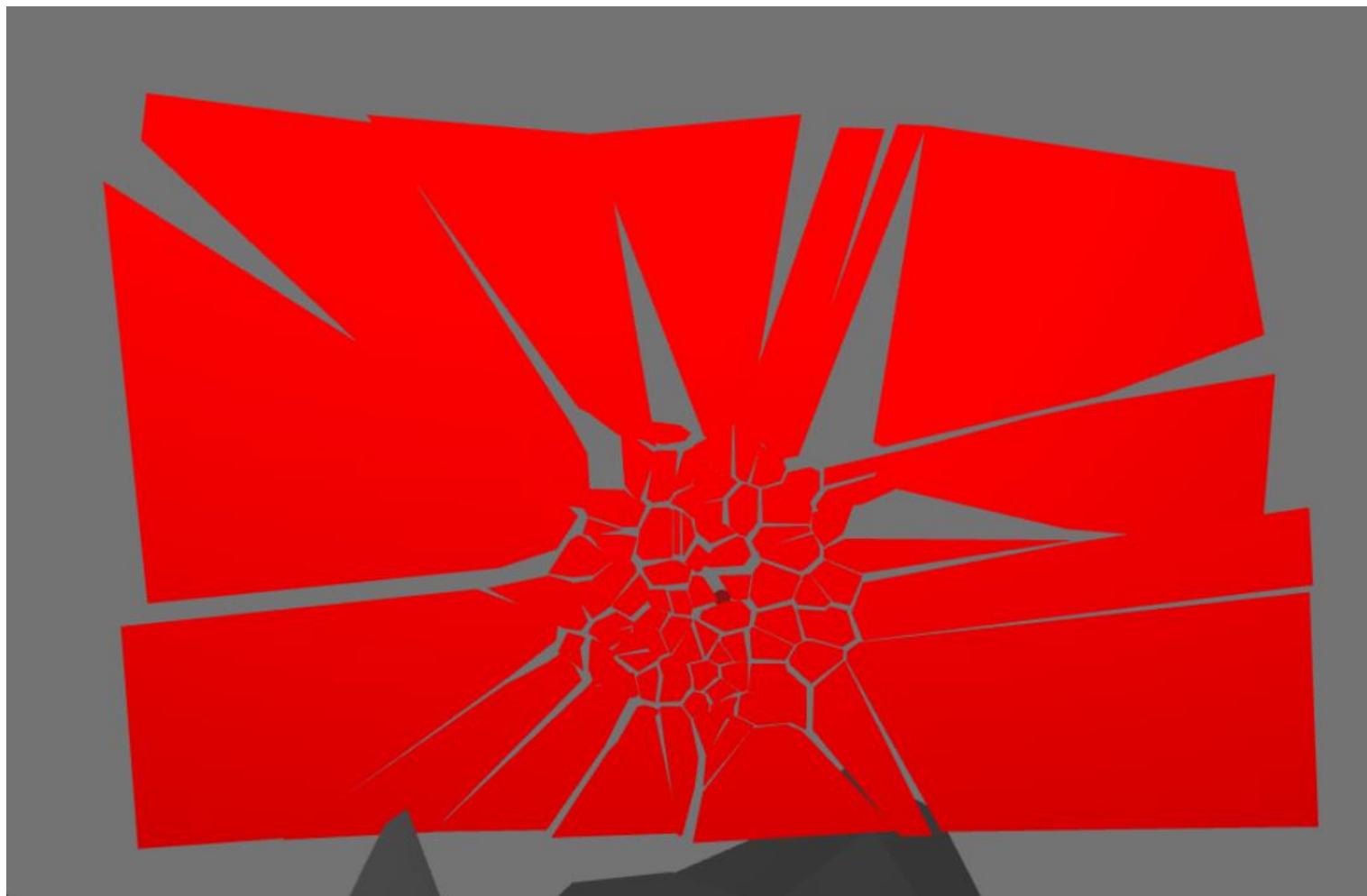
- BroadPhase
 - AABB Tree
- NarrowPhase
 - SAT(Separating axis theorem)
- Solveur itératif
 - Vitesse
 - Position



Démonstration



Destruction de maillages en 2 dimensions



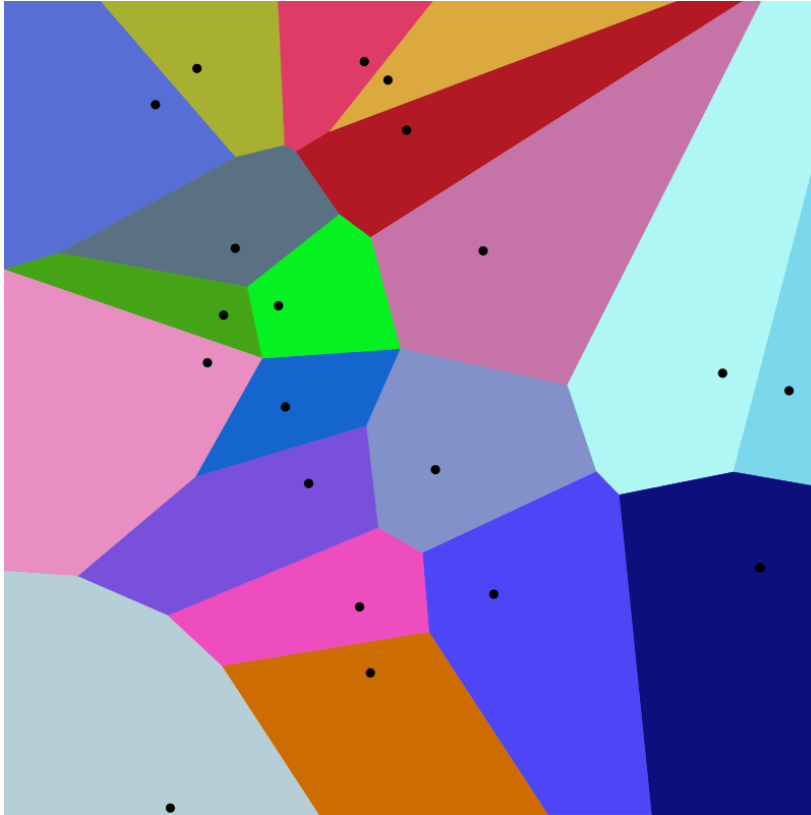
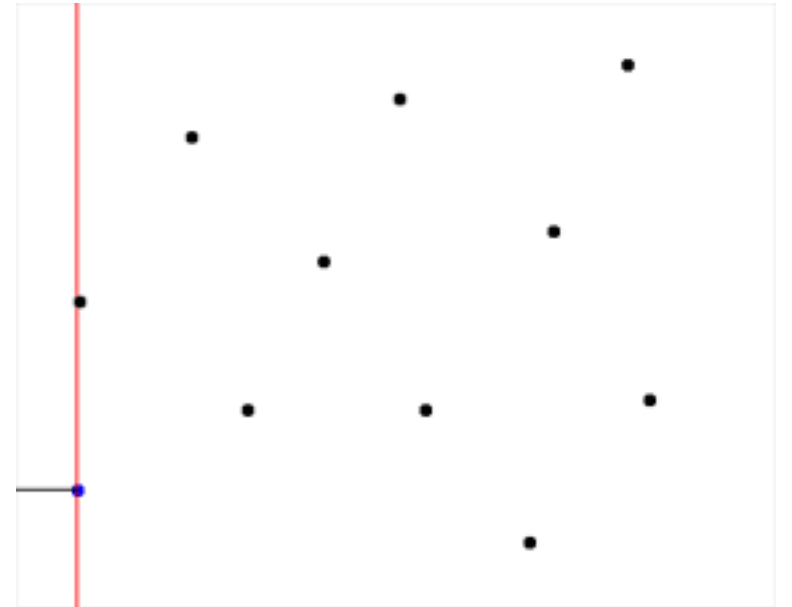


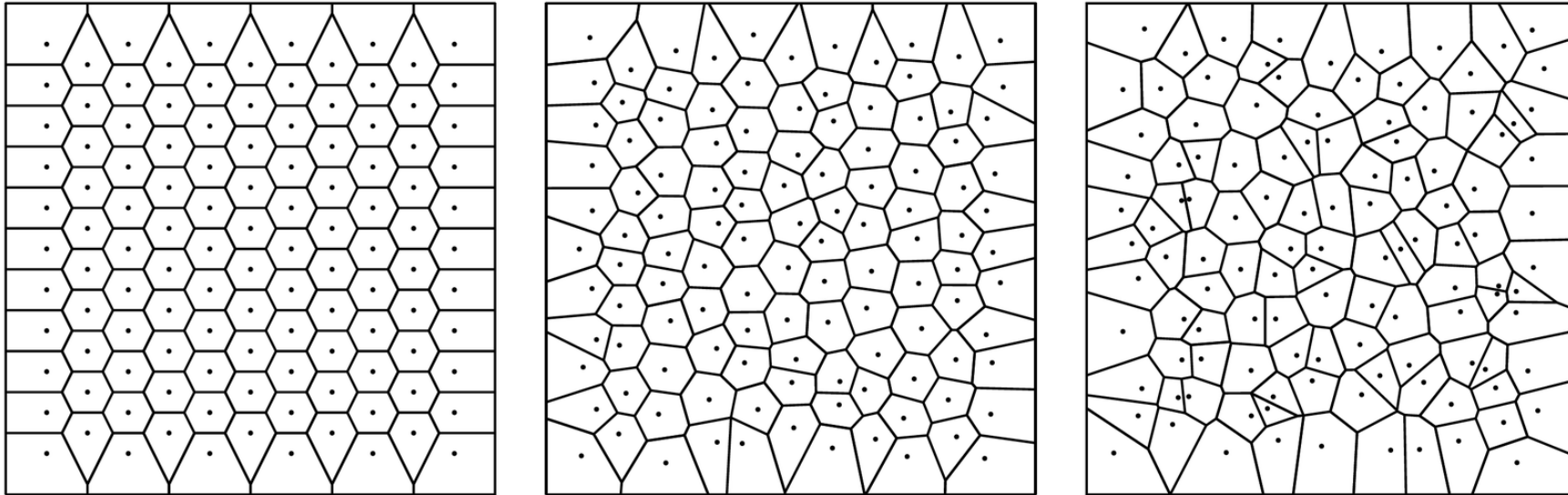
Diagramme de Voronoï en 2 dimension

- Une méthode de pavage de plan
- Chaque pavé correspond à l'ensemble des points du plan le plus de la seed, pour chaque seed.
- Selon l'ensemble de point de départ (seeds), peut offrir un bel effet de fragmentation

Algorithme de Fortune

- Contrainte temps réel
- Utilisation de l'algorithme de Fortune
- Complexité temporelle $O(n \log n)$





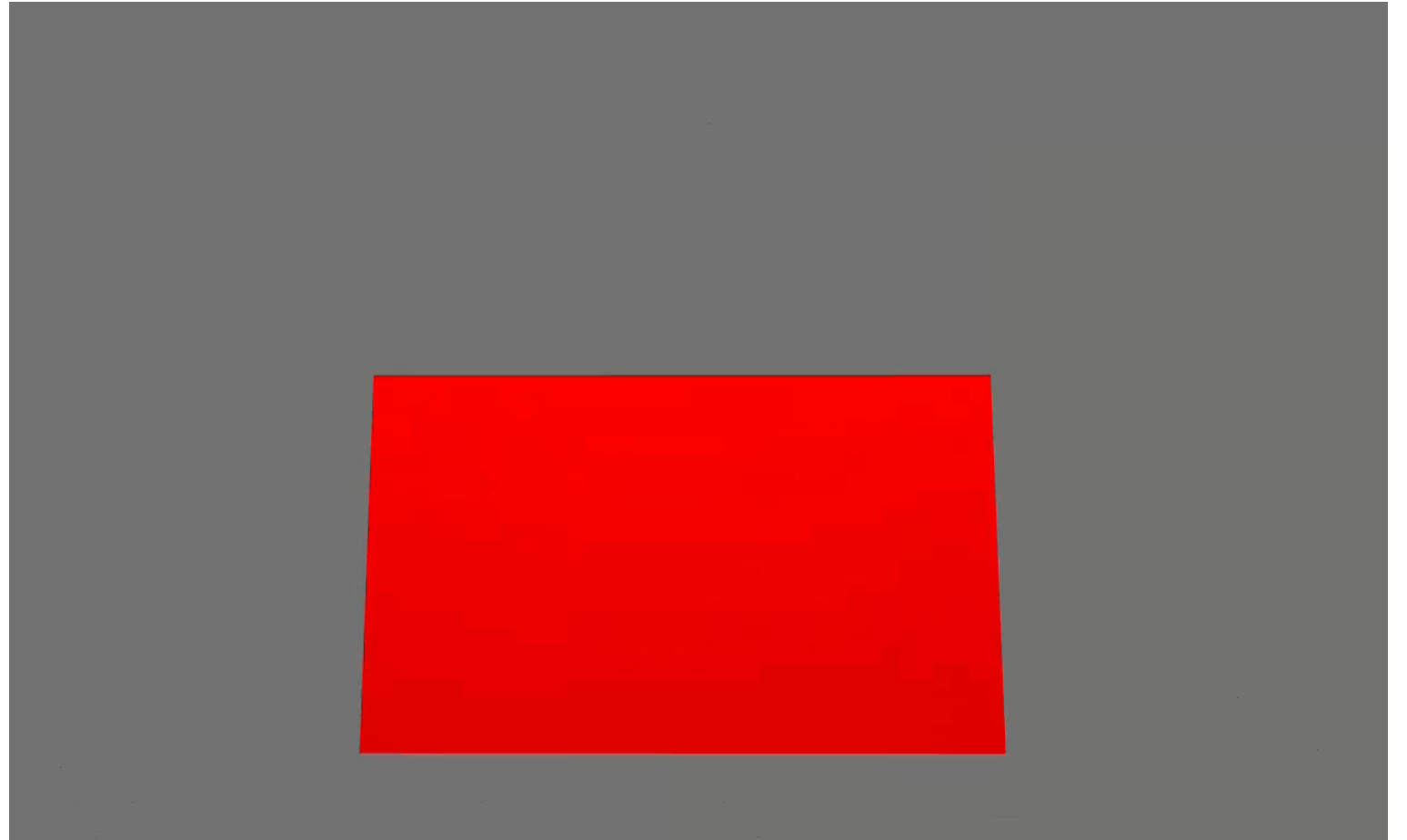
- Génère un pavage différent selon la génération des points
- Permet un contrôle de la fragmentation

Génération de points

Création d'entités à partir du diagramme

- Pour chaque cellule l'on crée une nouvelle entité destructible que l'on extrude du diagramme
- Une entité destructible est définie par :
 - Un polygone
 - Une profondeur de destruction
 - Une durée de vie

Démonstration



Implémentation future

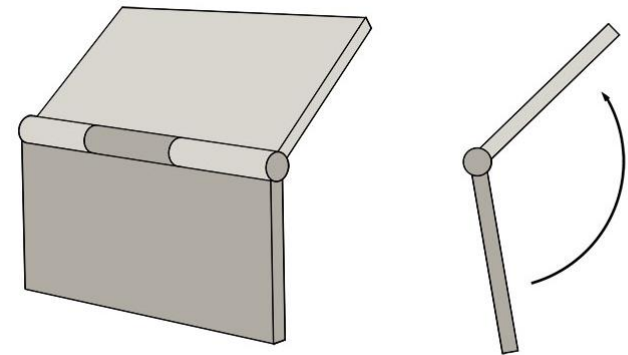
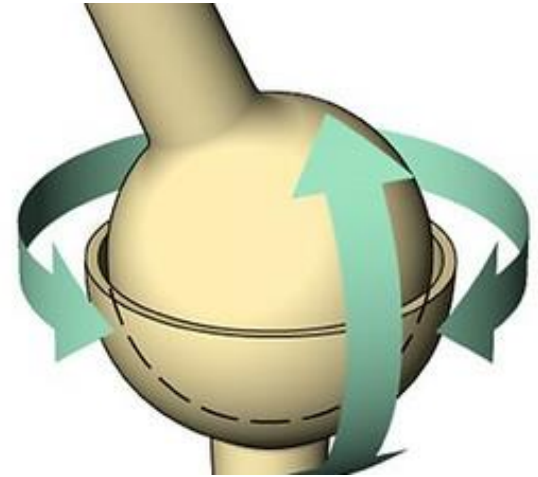
- Fonctionnalité pour le rendu
- Utilisation de joints
- Destruction d'objet en 3D

Fonctionnalité pour le rendu

- Gestion des ombres
- Meilleure gestion des matériaux

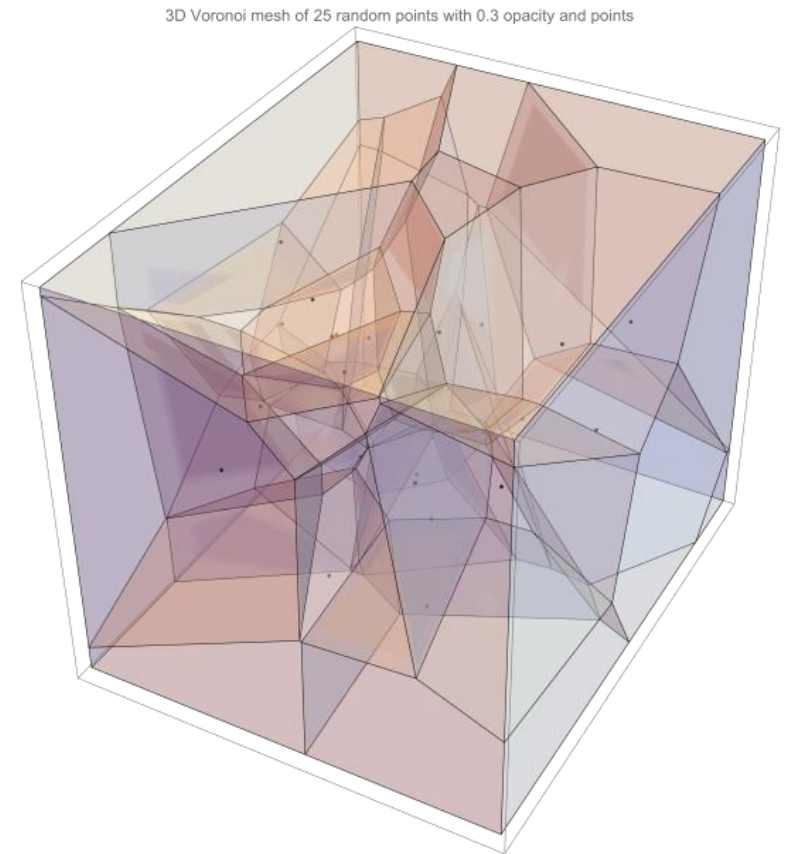
Utilisation de joints

- ReactPhysics3D
 - Ball and Socket(3 degrés de liberté rotation)
 - Hinge(1 degré de liberté rotation)
 - Slider(1 degré de liberté translation)
 - Fixed(0 degré de liberté)



Destruction d'objets en 3 dimensions

- Génération de diagramme de Voronoï en temps réel difficile
- Pré-traitement possible



Références

- Images :
 - https://commons.wikimedia.org/wiki/File:Euclidean_Voronoi_diagram.svg
 - <https://commons.wikimedia.org/wiki/File:Fortunes-algorithm-slowed.gif>
 - https://en.m.wikipedia.org/wiki/File:3D_Voronoi_mesh_of_25_random_points_with_0.3_opacity_and_points.svg
 - <https://leswordfish.com/portfolio/the-separating-axis-theorem/>
 - <https://allenchou.net/2014/02/game-physics-broadphase-dynamic-aabb-tree/>
 - <https://www.wolframscience.com/nks/notes-7-8--voronoi-diagrams/>
- Sources :
 - <https://www.reactphysics3d.com/documentation/>
 - <https://jerryronnegren.myportfolio.com/real-time-dynamic-destruction>
- Codes :
 - <https://github.com/Patateon/Moteur-de-jeux>

A 3D rendered white cartoon character with a large head, a simple smile, and thin limbs. The character is holding a large, rectangular white sign with both hands. The sign has a thin black border and the word "Merci" is written on it in a bold, black, cursive script. The character is standing on a light-colored surface, and the background is a soft, light blue gradient.

Merci