Animation par modèle physique

Florent Diet
Juin 2022

1 Animation d'un objet

Ici nous cherchons a simuler un drap en 3D. Comme le drap n'est pas un objet rigide nous ne pouvons pas nous permettre de ne calculer que le centre de masse et définir l'objet avec ceci c'est pourquoi nous avons un maillage 70*70 qui représente le drap et qui va nous permettre de simuler les ondulations. Le but sera donc de suivre la trajectoire de toutes ces particules en fonctions des forces et contraintes qu'elle subissent au cour du temps afin d'avoir une simulation correcte. Tout d'abord pour chaque pas de temps, nous devons implémenter les actions suivantes pour chaques particules :

- Calcul des forces de gravité
- Calcul des forces d'un systeme masse ressort
- Calcul de l'accélération
- intégration de la vitesse
- intégration de la position
- mise a jour des positions

Pour ce qui est de la gravité c'est assez simple cela correspond à

$$F_g = -m * g$$

Nous multiplions par -1 car nous appliquons cette force au sol, et ensuite nous multiplions la masse de la particule par la constante de la gravité 9.87N.

Pour ce qui est du calcul des forces d'un systeme masse ressort nous verrons cela dans la prochaine section.

Pour le cas du calcul de l'accélération d'une particule à un temps donné nous avons appris que d'après le principe fondamentale de la dynamique :

$$F = m * acceleration$$

F correspond à toutes les forces qui sont exercés sur la particule. Dans notre cas il sagit de la somme de la force de gravité et des forces des masses ressorts. Si nous transposons les termes de l'équation afin de n'avoir que l'accélération nous obtenons :

$$acceleration = F/m$$

Après avoir notre accélération et connaissant le théorème de Taylor nous pouvons trouver les vitesses et les positions de nos particules. Dans ce projet j'ai principalement utilisé l'algorithme de Euler semi implicite car c'est celui qui me donnait les résultats les plus stables.

$$v(t+h) = v(t) + h * acceleration(t)$$
$$x(t+h) = x(t) + h * v(t+h)$$

Après cette étape je n'avais plus qu'a mettre a jour les positions de chacunes de mes particules en mettant à jour les vertex qui y étaient lié. Voila le résultat finale :

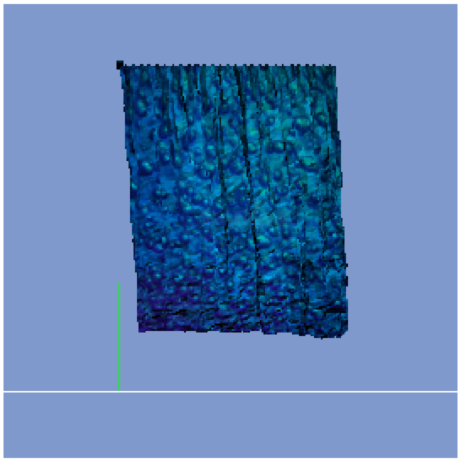

FIGURE 1 – drap de 70 * 70 particule de base

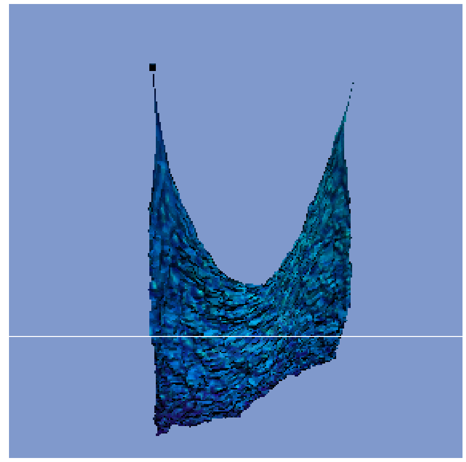

FIGURE 2 – drap qui subit les forces extérieurs

2 système masses-ressorts

Relier les particules par des "ressorts" nous permet de simuler les déformations du textiles Un système masse ressort est utilisé quand en imagerie on veut représenter un objets déformables. Le principes est de créer un maillaiges de cette objets avec des particules et des ressorts se trouvant entre ces particules. Ces ressorts permettront suivant les forces appliqués sur ceux ci de simuler la déformation de l'objet.

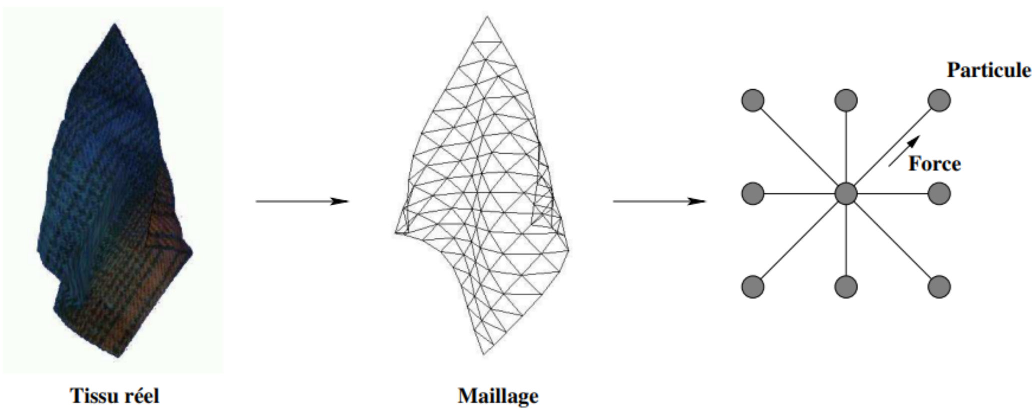

FIGURE 3 – exemple systeme masse ressort

Pour calculer ses forces entre ces ressorts nous utilisons la loi de Hooke :

$$F = -k\Delta u$$

Ou la force est toujour opposé a la deformation c'est pour cela que l'on multiplie le tout par -1. Δ correspond au déplacement du ressort et k est la constante de reaideur du ressort.

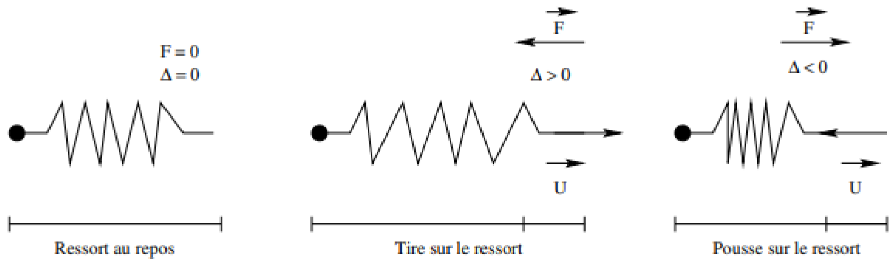

FIGURE 4 – Aspect visuel de la loi de Hooke

En approfondissant un peu plus cette loi nous arrivons à cette équation qui correspond à l'equation finale utilisé dans ce projet :

$$F_{i,j}(t) = -k_{i,j} * (||x_i(t) - x_j(t)|| - l_{i,j}) * u_{i,j}$$

3 Evolution

3.1 Collision

J'ai tout d'abord ajouté une collision au sol en disant que si jamais une particule a sa coordonnée y qui est au niveau du sol alors sa vitesse en y est automatiquement de 0. J'ai aussi ajouté une collision avec une sphère. Pour ce faire j'ai tout d'abord utilisé les différents objets qui étaient laissé dans le dossier data pour les insérer dans mon environnement après avoir récupéré le centre et le rayon de cette sphère j'ai calculé la distance entre le centre de ma sphere et les particules et si jamais la distance était égal ou inférieur au rayon sa vitesse en y est mise à 0.

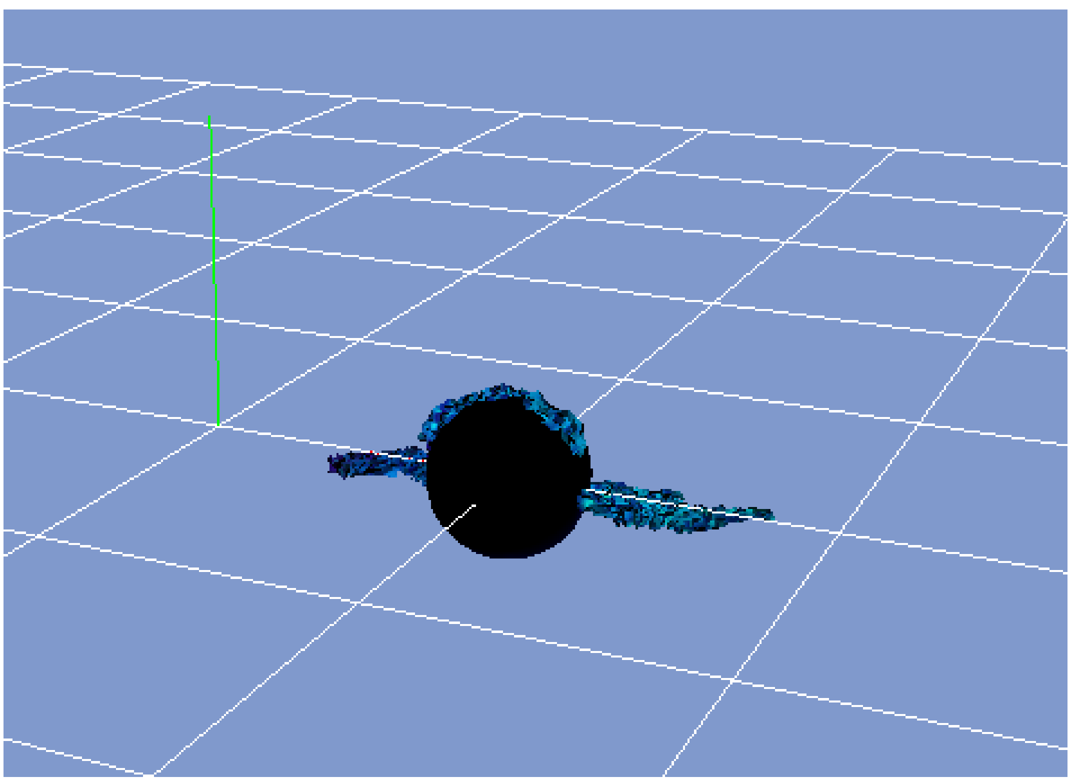

FIGURE 5 – Exemple de collision avec le sol et la sphère

3.2 Mouvement d'un coin

J'ai tout simplement ajouter aux coordonnées du point correspondant au coin gauche la possibilité de se déplacer en ajoutant un vecteur unitaire. Ce vecteur est initalisé suivant la direction dans laquel l'utilisateur a choisi de diriger le coin.

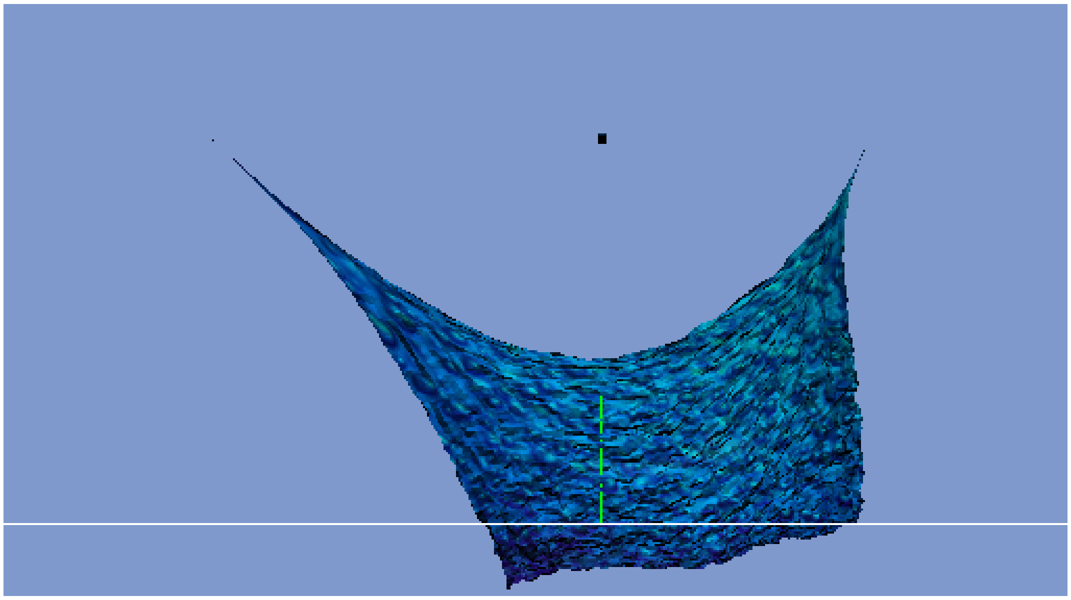

FIGURE 6 – Exemple de mouvement d'un coin

3.3 Déchirure

Pour faire une déchirure j'ai mesuré la force qu'exerce le systeme masse ressort d'une particule et si jamais cette force excede un certain montant alors sa raideur est mise a 0 ce qui va supprimer le ressort en question.

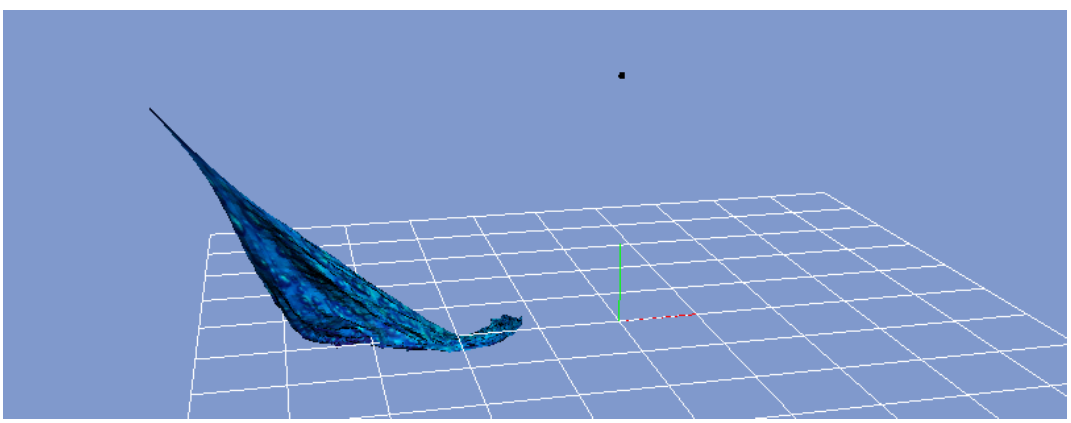

FIGURE 7 – Exemple de déchirure

4 futur

Si je disposais de plus d'heure pour améliorer ce projet il aurait été interessant d'étudier les forces de frottement car l'on voit par exemple que dans le cas des collisions je pense qu'il serait possible de rendre les résultat plus réaliste. Aussi pour ce qui est de la déchirure j'aurais aimé pouvoir l'implémenter de sorte à ce que la dechirure ce fasse progressivement, ici tout s'arrache d'un coup. La comparaison avec l'animation d'un vrai drap ou je pourrais faire une étude de ses animations pourrais aussi s'avérer interessant afin de comparer nos résultat. Aussi nous aurions pu reproduire ces phénomènes sur des objets beaucoup plus gros et avec des matières différentes donc des constantes de raideur différentes afin de regarder les réactions qu'ils penvent avoir avec l'environnement. En agrandissant les objets nous aurions pu essayer de paralelliser le code.