Journal de bord : Phase 2 (Groupe7 - Pac Man)

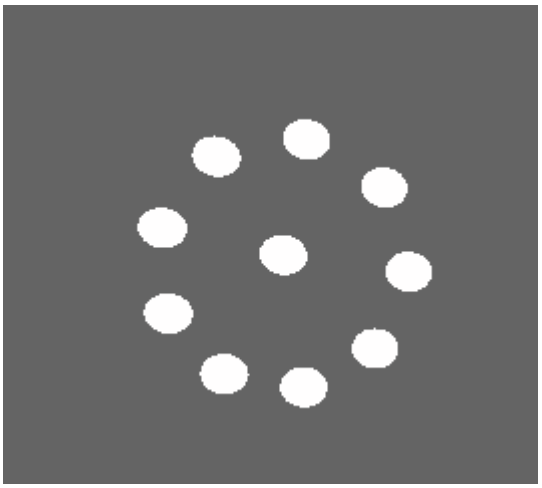
Introduction :

L'objectif de notre deuxième phase est de pouvoir faire se déplacer un blop sur le sol. Ce blop doit être un amas de particule tel que :

- Les particules doivent détecter aux collisions et les résoudre avec des impulsions
- Chaque particule subit des forces, comme la gravité ou la friction, et peuvent subir des forces individuellement selon les objets qu'elles rencontrent
- Chaque particule du blop doit être liée à ses voisines par un ressort invisible, sauf la particule du milieu qui est liée à toutes les autres.
- En bonus, le blop peut se déplacer a gauche ou a droite, fusionner ses particules ou les diviser

Démonstration :

Le blop :

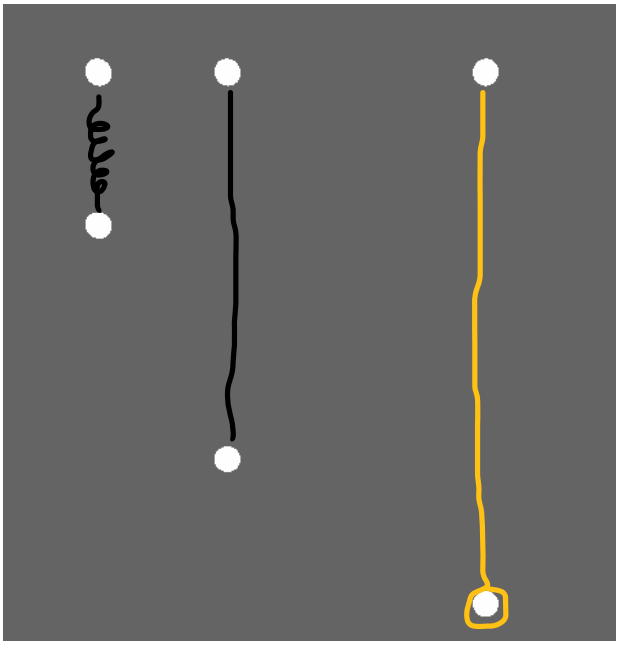


Le blop est formé de particules reliées par des ressorts (invisibles) à leurs voisines et à la particule du milieu.

Il peut se déplacer vers la droite ou vers la gauche, et peut s'envoler, mais chacune de ses particules reste soumise à la gravité donc il retombera au sol.

On peut voir la gravité dès le lancement du programme, puisque les particules du blob ainsi que les particules de démonstration des collisions tombent vers le sol naturellement.

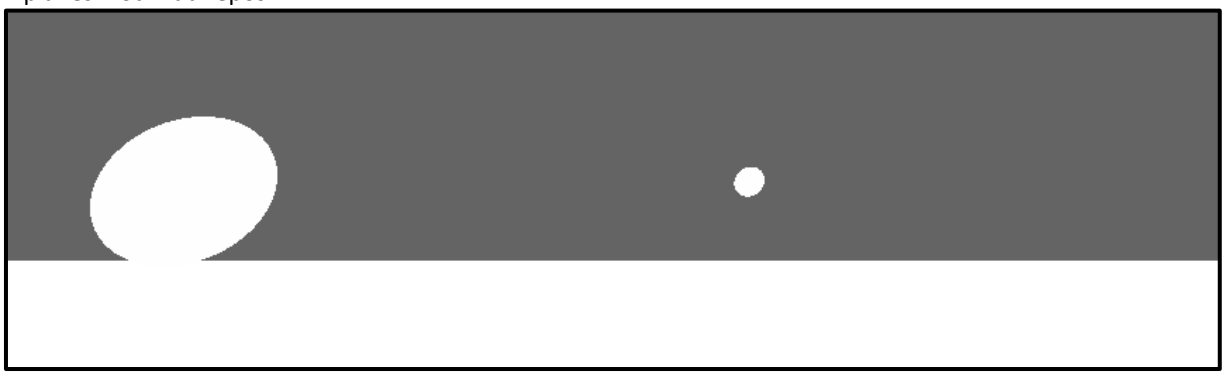
Exemple des différents types de ressort (invisibles dans le programme) :



De gauche à droite :

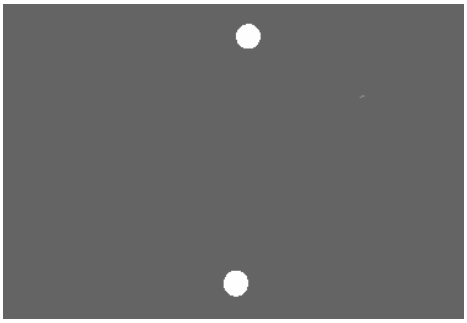
- Un ressort où la particule est attachée à une longueur supérieure à la longueur au repos, ce qui provoque de légères oscillations puis l'équilibre est atteint.
- Un câble où la une fois l'élasticité maximale est dépassée, la particule est stoppée dans sa chute et retenue par le câble.
- Un élastique, qui fonctionne comme un ressort mais uniquement lorsqu'il est allongé, la compression ne provoque pas de force de rappel

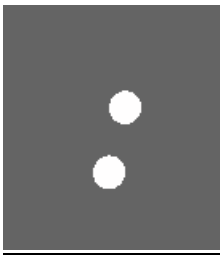
Il existe aussi un sol dans la scène sur lequel les particules peuvent rebondir, on parle de résolution de collisions « planes » ou « au repos »

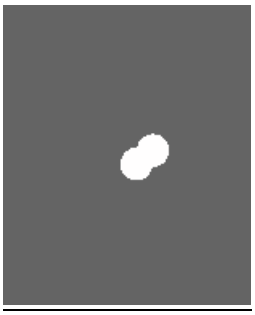


Exemples de collision :

Voici plusieurs frames qui suivent le parcours de deux particules qui s’entrechoquent :







Explication du code :

Pour des raisons de lisibilité, nous n'avons pas recopié ici le code car le rapport était surchargé.

La classe RegistreForce :

Elle est responsable de la gestion et de l'application des forces sur les particules du système. La méthode add permet d'ajouter une force générée par un GenerateurForce spécifique à une Particule. Cela permet de spécifier quelles forces affectent quelles particules. La méthode clear supprime toutes les forces enregistrées dans le registre, ce qui est utile pour réinitialiser le système. La méthode remove permet de retirer une force spécifique associée à une particule donnée, offrant ainsi un moyen de supprimer des forces individuelles du registre. La méthode updateForces parcourt toutes les forces enregistrées dans le registre et utilise les générateurs de force pour mettre à jour les forces appliquées sur les particules en fonction de la durée d'une frame (indiquée par duration).

Toutes les classes qui simulent une force héritent d'une classe GenerateurForce et implémentent donc sa méthode updateForce. Nous avons plusieurs classes forces comme ForceRessort, ForceFrictionCinétique… Nous allons prendre l'exemple de ForceRessort :

La classe ForceRessort permet de modéliser un ressort. Elle prend en paramètres la constante du ressort (k), la longueur de repos du ressort (l0), le seuil d'élasticité (limiteElasticite), la position d'origine de la masse fixe (origine) *ou alors les deux particules reliées par le ressort* (*particule1, particule2)* , et le coefficient d'amortissement (coefficientAmortissement). La méthode updateForce calcule la force exercée par le ressort sur une particule donnée. Elle vérifie si la déformation du ressort dépasse le seuil d'élasticité, auquel cas elle applique une force supplémentaire pour empêcher une extension excessive et réduire la vitesse. Sinon, elle calcule la force du ressort et ajoute également une force d'amortissement en fonction du coefficient d'amortissement.

Commentaire : Pour cette phase 2, nous avons essayé de montrer sur la scène un maximum des différentes exigences, ce qui peut donner une scène assez chargée, c'est pourquoi nous vous conseillons de vous pencher 1 par 1 sur les différents éléments de l'écran pour vérifier le fonctionnement d'une implémentation en particulier.

Enfin, nous avons commencé à la suite de vos conseils à implémenter certains tests unitaires de Vecteur3D que vous trouverez dans la classe UnitTest. Malheureusement, celle-ci est encore légèrement incomplète car nous nous sommes davantage concentrés sur l'implémentations des forces physiques.