Simulation physique de corps rigides avec interaction

Merwan Achibet

Moteur physique ?

Moteur physique : système de simulation mécanique

Industrie, science, cinéma précis, coûteux

Jeu vidéo, réalité virtuelle approximatifs, temps réel

Ce projet :

- Moteur physique de base
- Corps rigides
- Corps convexes
- Temps réel

Étude de cas

1. La chute

$$\vec{a} = \frac{1}{m} \sum_i \vec{F}_i$$

2. Le rebond

$$\vec{v}_1 = \gamma \vec{v}_2$$

3. Le repos

$$\vec{F}_{A/B} = -\vec{F}_{B/A}$$
$$\vec{F}_{A/B} + \vec{F}_{B/A} = 0$$

Modules

Différentes tâches :

- Dynamique
  - Mouvement linéaire
  - Mouvement angulaire
- Gestion des collisions
  - Détection
  - Correction
  - Réponse

La composante linéaire

- Entrée : forces environnementales
- Sortie : changement de position

$$\vec{v} = \frac{\partial \vec{p}}{\partial t}$$
$$\vec{a} = \frac{\partial \vec{v}}{\partial t}$$

$$\vec{p} = \int \vec{v}\,\partial t$$
$$\vec{v} = \int \vec{a}\,\partial t$$

Intégration approximée

Intégration d'Euler :

$$x_{n+1} = x_n + x' \partial t$$

Appliquée à nos besoins :

$$\vec{a}_{n+1} = \frac{1}{m} \sum_i \vec{F}_i$$

$$\vec{v}_{n+1} = \vec{v}_n + \vec{a}_{n+1} \partial t$$

$$\vec{p}_{n+1} = \vec{p}_n + \vec{v}_{n+1} \partial t$$

Simplification grâce à l'élan linéaire

$$\sum_i \vec{F}_i = \frac{\partial \vec{L}}{\partial t} = \frac{\partial (m\vec{v})}{\partial t}$$

$$\vec{L}_{n+1} = \vec{L}_n + \sum_i \vec{F}_i$$

$$\vec{p}_{n+1} = \vec{p}_n + \frac{1}{m}\vec{L}_{n+1}\partial t$$

Modélisation d'un corps

OK pour une particule, mais un objet plus complexe ?

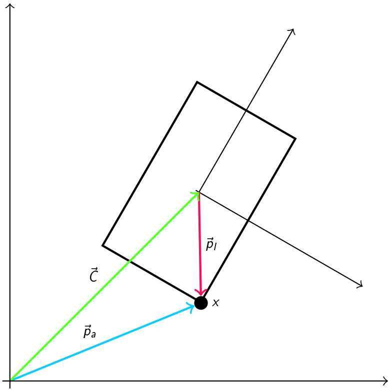
Une particule = un sommet Non

Une unique particule judicieusement placée Oui, le centre de masse

$$\vec{C} = \frac{1}{M} \sum_i m_i \vec{p}_i$$

Le centre de masse

Centre de masse = origine du repère local



$$\vec{p}_l = \vec{p}_a - \vec{C}$$

La composante angulaire

Il manque quelque chose... Les rotations !

Orientation Une matrice : un vecteur colonne = un axe du repère local

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Élan angulaire Analogue à l'élan linéaire

$$\vec{A}_{n+1} = \vec{A}_n + \sum_i \vec{\tau}_i$$

$$\vec{\tau}_i = (\vec{x} - \vec{C}) \times \vec{F}_i$$

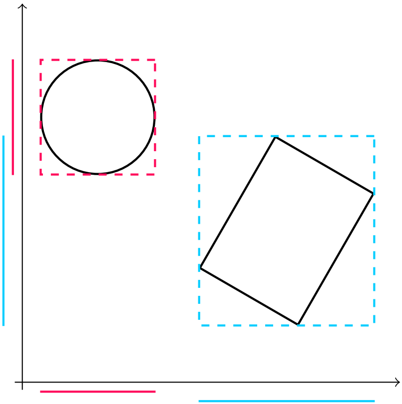
quantités auxiliaires

Deux niveaux de précision

On teste les collisions entre paires de corps : $\frac{n(n-1)}{2}$ tests

1. Détection grossière
2. Détection fine

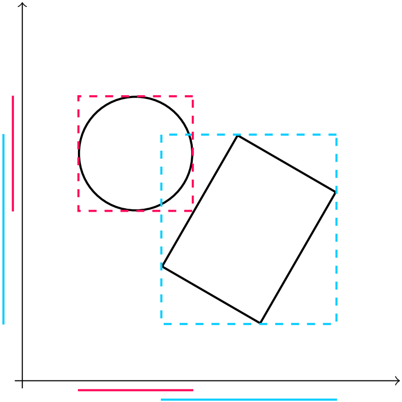
Détection grossière I



Boîte englobante Contient tous les points d'un corps

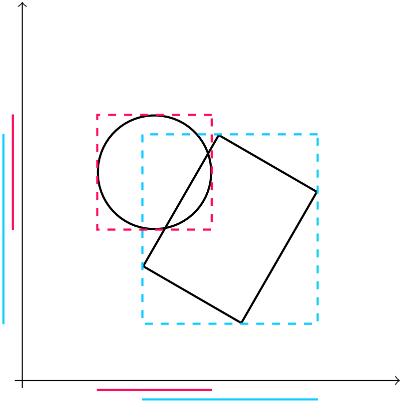
SAT Test de collision rapide

Détection grossière II



- Résultat positif quand deux corps s'interpénètrent, mais aussi parfois quand ce n'est pas le cas.
- La détection fine validera/infirmera la collision

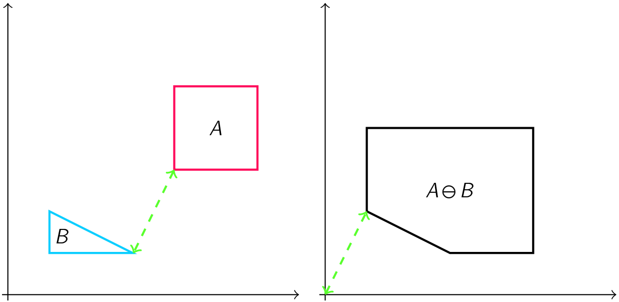
Détection grossière III



Détection fine I

Somme de Minkowski $A \oplus B = \{a + b \mid a \in A, b \in B\}$

Différence de Minkowski $A \ominus B = A \oplus (-B)$



Particularité la plus petite distance de la différence de Minkowski à l'origine est la plus petite distance entre les corps A et B

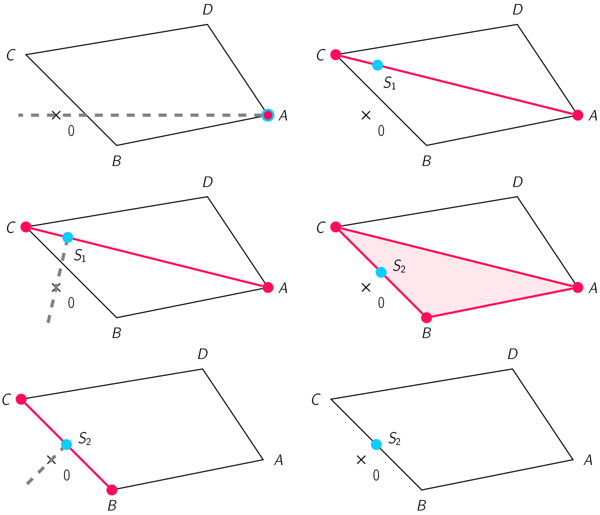
Détection fine II

Comment calculer la plus petite distance entre M et l'origine ?
Un simplex :

1. Sommet
2. Arête
3. Triangle
4. Tétraèdre

Une fonction de support :

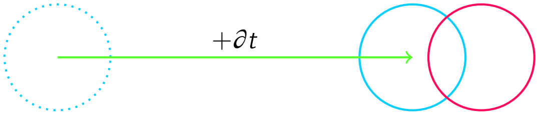
détection fine III



Correction I

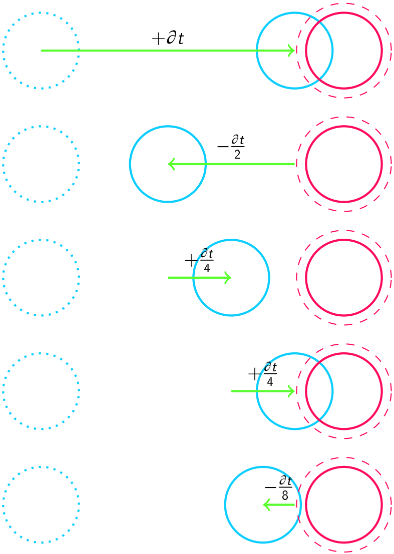
Simulation discrète Pas de temps fixe

Problème Les collisions sont toujours pénétrantes



Correction II

Solution Intégrer en arrière, par dichotomie

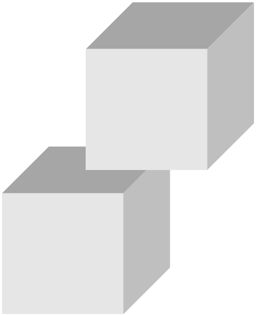


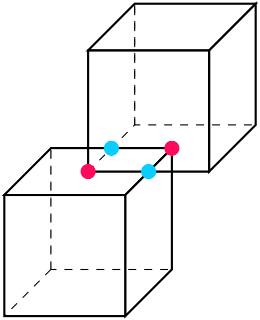
Réponse I

Un corps rigide :

- Sommets
- Arêtes
- Faces

On s'intéresse uniquement aux contacts sommet-face et arête-arête.





Réponse II

Un *contact* :

- Position
- Normale
- Temps

Un contact = une *impulsion*

$$J = \vec{n}\frac{-(1+\varepsilon)v_r}{\frac{1}{m_A}+\frac{1}{m_B}+\vec{n}(I_A^{-1}(\vec{r}_A\times\vec{n}))\times\vec{r}_A+(I_B^{-1}(\vec{r}_b\times\vec{n}))\times\vec{r}_B}$$

Algorithme

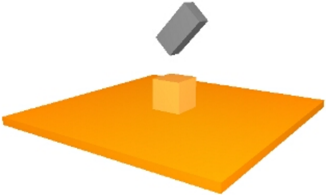
1. Détection de collision
   1.1 Détection grossière
   1.2 Détection fine
   1.3 Recherche des contacts
   1.4 Application d'impulsions
2. Application de forces environnementales
3. intégration

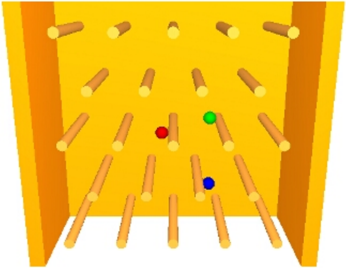
défauts

Algorithme amélioré

mieux

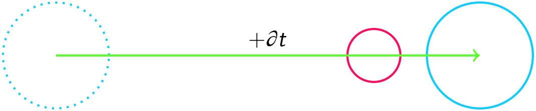
Démonstrations





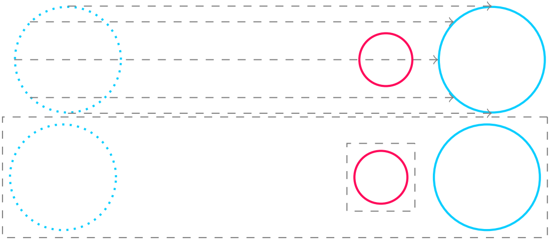
Tunneling I

Toujours à cause de l'intégration discrète :

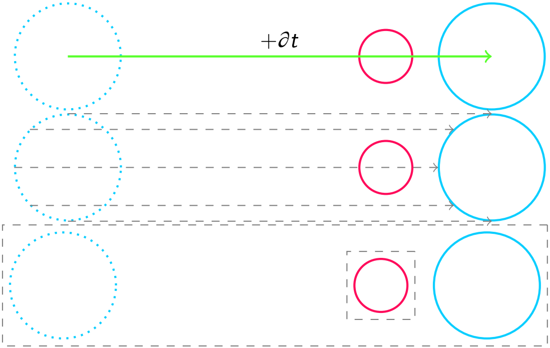


Les corps se traversent mais aucune collision n'est détectée !

Tunneling II



Partitionnement de l'espace



Conclusion