

Session 6 : Broad & Narrow Phases

1. Le Contexte : Le Budget Temps (16ms)

Dans un jeu vidéo tournant à 60 FPS, vous avez 16.6 millisecondes pour **tout** calculer.

La physique n'est qu'une part du gâteau :

- Rendu (Graphics) : 8ms
- IA / Gameplay : 4ms
- Audio / Réseau : 2ms
- **Physique : 2ms**

Si votre détection de collision prend 10ms, le jeu saccade (« lag »). L'optimisation n'est pas un bonus, c'est une nécessité vitale.

2. Le Problème Mathématique : La Malédiction O(N²)

Pourquoi la détection « naïve » échoue-t-elle ? Imaginons une matrice d'interaction où chaque objet (Ligne) teste chaque objet (Colonne).

Pour $N = 5$ objets (A, B, C, D, E) :

	A	B	C	D	E
A	X	Test	Test	Test	Test
B	Skip	X	Test	Test	Test
C	Skip	Skip	X	Test	Test
D	Skip	Skip	Skip	X	Test
E	Skip	Skip	Skip	Skip	X

- La Diagonale (X) : Test inutile (A vs A).
- Le triangle inférieur (Skip) : Redondant (si A touche B, B touche A).

Le Coût Réel :

$$C \approx \frac{N(N - 1)}{2} \approx O(N^2)$$

- 10 objets → 45 tests (Négligeable)
- 1 000 objets → 499 500 tests (Lent)
- 10 000 objets → 50 000 000 tests (Crash CPU)

Solution : Il faut réduire N avant de faire les tests coûteux. C'est le rôle de la Broad Phase.

3. La Broad Phase : Structures de Partitionnement

Le but est de répondre rapidement à la question : « **Qui sont mes voisins ?** ». On remplace les objets complexes par des AABB (Axis Aligned Bounding Box).

A. Grille Uniforme (Spatial Hashing)

- Concept : On découpe le monde en cases de taille fixe. On ne teste l'objet A que contre les objets dans la même case (ou les cases adjacentes).
- Avantage : Accès $O(1)$, très rapide, facile à coder.
- Inconvénient : Mauvais si le monde est infini ou si les objets ont des tailles très variées.

B. Arbres Spatiaux (Quadtree / Octree)

- Concept : Découpage récursif. Si une case contient trop d'objets, on la coupe en 4 (2D) ou 8 (3D).
- Avantage : S'adapte à la densité (beaucoup de cases là où il y a des objets, peu ailleurs).
- Inconvénient : Coûteux à mettre à jour si les objets bougent beaucoup.

C. Sweep and Prune (SAP)

- Concept : On projette les débuts/fins des AABB sur les axes X, Y, Z et on trie la liste. Si les intervalles sur X ne se chevauchent pas, inutile de tester Y et Z.
- Secret : Exploite la **Cohérence Temporelle** (les objets ne se téléportent pas, la liste reste presque triée d'une frame à l'autre).

4. La Narrow Phase : Les Méthodes de Résolution

On a filtré 99% des cas. Il reste les paires « proches ». Comment savoir si elles se touchent vraiment ? Deux écoles dominent :

Méthode 1 : SAT (Separating Axis Theorem)

- Philosophie : « L'ombre ». Si je peux trouver une lumière qui projette des ombres séparées des deux objets, alors ils ne se touchent pas.
- Fonctionnement : On projette les formes sur une série d'axes (Normales des faces, Produits vectoriels des arêtes). Si **une seule** projection montre un écart, il n'y a pas collision.
- Utilisation : Idéal pour Boîtes vs Boîtes ou Polyèdres simples.

Méthode 2 : GJK (Gilbert-Johnson-Keerthi)

- Philosophie : « La Différence ». On travaille dans l'espace de Minkowski ($A - B$).
- Fonctionnement : On cherche itérativement si l'Origine $(0, 0, 0)$ est contenue dans la forme $A - B$ en construisant un **Simplex** (Tétraèdre) à l'intérieur.
- Utilisation : Le standard pour les formes convexes quelconques (Capsules, Cones, Mesh convexes).

Focus : Comprendre GJK et le Support Mapping

GJK est un algorithme « aveugle ». Il ne connaît pas la géométrie, il pose juste une question à l'objet via une Fonction de Support :

« **Quel est ton point le plus extrême dans la direction \vec{d} ?** »

$$S_{A-B}(\vec{d}) = S_{A(\vec{d})} - S_{B(-\vec{d})}$$

Cela permet de découpler l'algorithme de collision (GJK) de la définition géométrique des objets.

- Sphère : Centre + Rayon * \vec{d}
- Cube : Coin correspondant au signe de \vec{d}

Résumé du Pipeline

1. **Update Position :** $P = P + V * dt$
2. **Broad Phase :** Mise à jour de la grille/Octree -> Liste de paires candidates.
3. **Narrow Phase :**
 - Check AABB (rapide)
 - Check GJK ou SAT (précis) -> Collision OUI/NON
4. **Contact Generation (EPA/Clipping) :** Point de contact, Normale, Profondeur.
5. **Resolution :** Application des impulsions pour séparer les objets.