

Collision Detection

Milestones

Tag Version	Deadline
BroadPhase	25/11/2022 – 17h30
NarrowPhase	19/12/2022 - 17h30
Reponse	02/01/2022 - 17h30

Nombre d'étudiants par groupe : 1

Rendu :

- Fichiers : binaires (dans Collision_Detection\nom_du_groupe\BIN)
- Sources : dans Collision_Detection\nom_du_groupe\SOURCES
- Screenshots : 5 (dans Collision_Detection\nom_du_groupe\SCREENS)

Sujet

Vous devez repartir de la solution présente dans le dossier Support Cours/Moteur Physique/CollisionEngine, et la compléter de façon à obtenir le même résultat que l'exécutable dans le dossier Target (à l'intérieur du dossier CollisionEngine)

Plus précisément, vous devez implémenter les 2 phases de la détection de collision : la broadphase et la narrowphase suivi de la réponse. Vous êtes libres dans le choix de la technique utilisée pour chacune des phases, vous serez jugés sur le bon fonctionnement du programme (comportement attendu, pas de bugs), les performances et la lisibilité du code.

Guideline et Rendu

Pour vous mieux s'organiser et ne pas rendre tout au dernier moment, vous allez avoir des étapes de rendu (CF Milestones)

Broad Phase:

Aussi appelé phase de trie, vous devez implémenter un algorithme qui va trier les collisions possibles. Vous pouvez utiliser les algo du cours ou d'ailleurs (Circle to Circle, Sweep and prune, AABB tree ou le Grid appelé aussi partitionnement spatiale)

Narrow Phase:

C'est ici que vous allez coder le test de collision et en ressortir les informations sur les paires testées. Encore une fois, vous pouvez choisir la méthode (SAT, GJK, ...), je vous encourage cependant à explorer le GJK qui est très efficace!

Reponse:

Pendant cette étape vous devrez traiter les objets en fonction des informations obtenues pendant la narrow phase. (Appliquer des forces à l'aide du vecteur de pénétration par exemple)

Bien-sûr toute copie de code vaudra un 0!