ESIEE – E3FI - 2022-2023

Syllabus du cours intitulé « Mathématiques pour la 3D »

par David Bilemdjian

L'objectif de ce cours est d'explorer et mettre en œuvre certaines techniques mathématiques et certains algorithmes de la 3D.

Le développement d'un mini « moteur physique » de jeu vidéo 3D servira de fil rouge et d'objectif applicatif du cours.

Les développements seront effectués en langage C/C++ sous framework opensource *raylib* (www.raylib.com). Il n'est pas nécessaire de maîtriser les langages C/C++ pour aborder ce module, car un sous-ensemble extrêmement restreint des fonctionnalités des langages sera utilisé.

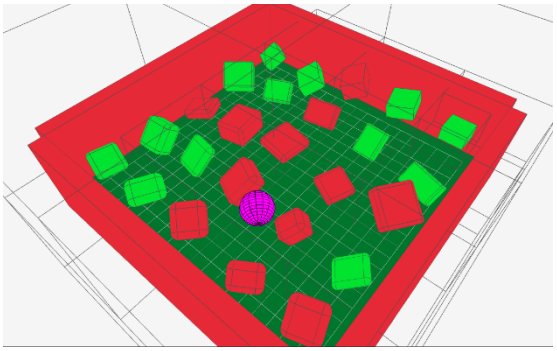


Figure Une boule rebondit sur des obstacles

La découverte de *raylib* sera conjointe de l'implémentation informatique des différents algorithmes.

Rappels de notions 3D

- Le référentiel 3D
- L'objet 3D : vertices et triangles
- Les différents systèmes de coordonnées en 3D : cartésien, cylindrique et sphérique
    - Méthodes de conversion
- Implémentation d'une caméra orbitale
- Méthodes de changement de référentiel

Rappels sur le produit scalaire : quantité de projection d'un vecteur sur un autre

Rappels sur le produit vectoriel

Introduction au paradigme d'orientation et de rotation « Angle-Axis » et aux quaternions

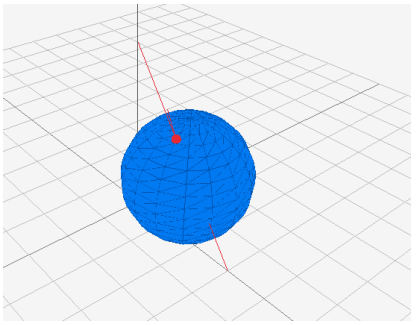
Concepts de détection de collision « *a priori* » versus « *a posteriori* ».

Détection de collision *a priori*

- Cas de la collision d'une sphère en mouvement de translation linéaire avec une box (parallélépipède)
    - Notion de « Sum de Minkowski »

Implémentation des algorithmes de calcul de l'intersection entre un segment et certaines primitives 3D

- Modélisation mathématique et graphique des primitives 3D : segment, plan, quad, oriented box, sphère, hémisphère, portion de sphère, disque, cylindre, capsule, rounded oriented box
    - Génération procédurale des objets: affichage des polygones et des wireframes sous openGL / raylib
- Intersection « segment – plan »
    - Puis « segment – quad »
- Intersection « segment – sphère »
- Intersection « segment – box (oriented box) »



- Intersection « segment – cylindre infini »
    - Projection d'un point sur une droite
- Intersection « segment – cylindre fini »
    - Intersection « segment - disque »
- Intersection « segment – capsule »
    - Distance d'un point à un segment
- Intersection « segment – rounded box »

Implémentation des algorithmes de détection de collision *a priori*

- Algorithme récursif de détection de collision entre une sphère en mouvement et plusieurs obstacles statiques (rounded oriented boxes)
    - Dynamique de la sphère : modèle simplifié de la translation et de la rotation de la sphère sous l'action de la force de gravitation et des forces de contact
    - Optimisation de la recherche spatiale par implémentation d'Octree (si le temps le permet)

Le cours est réparti du 14 septembre 2022 au 23 novembre 2022 ainsi:

- 5 cours magistraux de 3h
- 5 séances de TD de 3h

Livrables de fin de module : un projet Visual Studio 2019 sous framework *raylib* comprenant

- toutes les méthodes de détection de collision et d'intersection développées (le moteur physique)
- une scène de test, du type de celle présentée en cours où une boule rebondit de manière réaliste sur des obstacles statiques

Vous recevrez en temps et en heure un mail détaillant les modalités de livraison.

Les méthodes de détection d'intersection et de collision ainsi que le projet sont à développer en trinôme. Idéalement les trinômes doivent être équilibrés :

- Si la réflexion et la résolution mathématique doivent être menées individuellement, le développement peut être réparti
- Internet et la littérature peuvent être consultés sur les concepts de base, par contre je tiens à ce que vous élaboriez vos propres algorithmes,
- Interdiction de récupérer du code d'internet,
- Interdiction de récupérer du code d'un trinôme tiers, ou de personnes tierces internes ou externes à l'ESIEE,
- Vous pouvez éventuellement discuter avec les autres trinômes, mais ceux-ci ne sont autorisés qu'à vous mettre sur la voie. En aucun cas ils ne doivent vous donner une solution toute faite.

Vous l'aurez compris, je souhaite que vous réfléchissiez par vous-mêmes. Je préfère un code personnel pas forcément « académique » et peu optimisé qu'un code optimisé trouvé sur internet.