

Journal de bord

Phase 3 : implémentation des corps rigides dans notre moteur

I/ Répartition du travail préliminaire

Après avoir terminé la phase 2, nous avons suivi le cours sur les corps rigides et nous avons donc pu commencer à travailler sur cette phase. Nous avons écrit ensemble la déclaration de nos classes puis nous avons réparti l'implémentation :

- Eva travaillerait sur les classes Matrix33 et Matrix34
- Nicolas permettrait d'importer des modèles créés sur blender dans notre moteur graphique
- Nicolas travaillerait sur la classe Quaternion et sur le lien entre la matrice de transformation du moteur physique et celle du moteur graphique
- Célestin travaillerait sur la classe RigidBody

Aucun problème n'a été relevé à ce moment-là. Chacun a implémenté ce qu'il pouvait implémenter avec le cours que nous avons déjà eu, et nous avons testé la validité des opérations sur les matrices et sur les quaternions.

La classe RigidBody n'avait pas encore énormément de contenu à ce stade là, et nous n'avons pas encore compris la manière dont nous allions nous servir de la matrice de transformation.

Nicolas a réussi à importer n'importe quel modèle créé sur blender dans notre moteur graphique. Nous avons donc sélectionné deux modèles de voitures pour effectuer nos tests par la suite.

Le moteur graphique ne pouvait pas encore récupérer la matrice de transformation et nous n'avons donc pas encore pu tester le lancement d'un corps rigide avec une rotation.

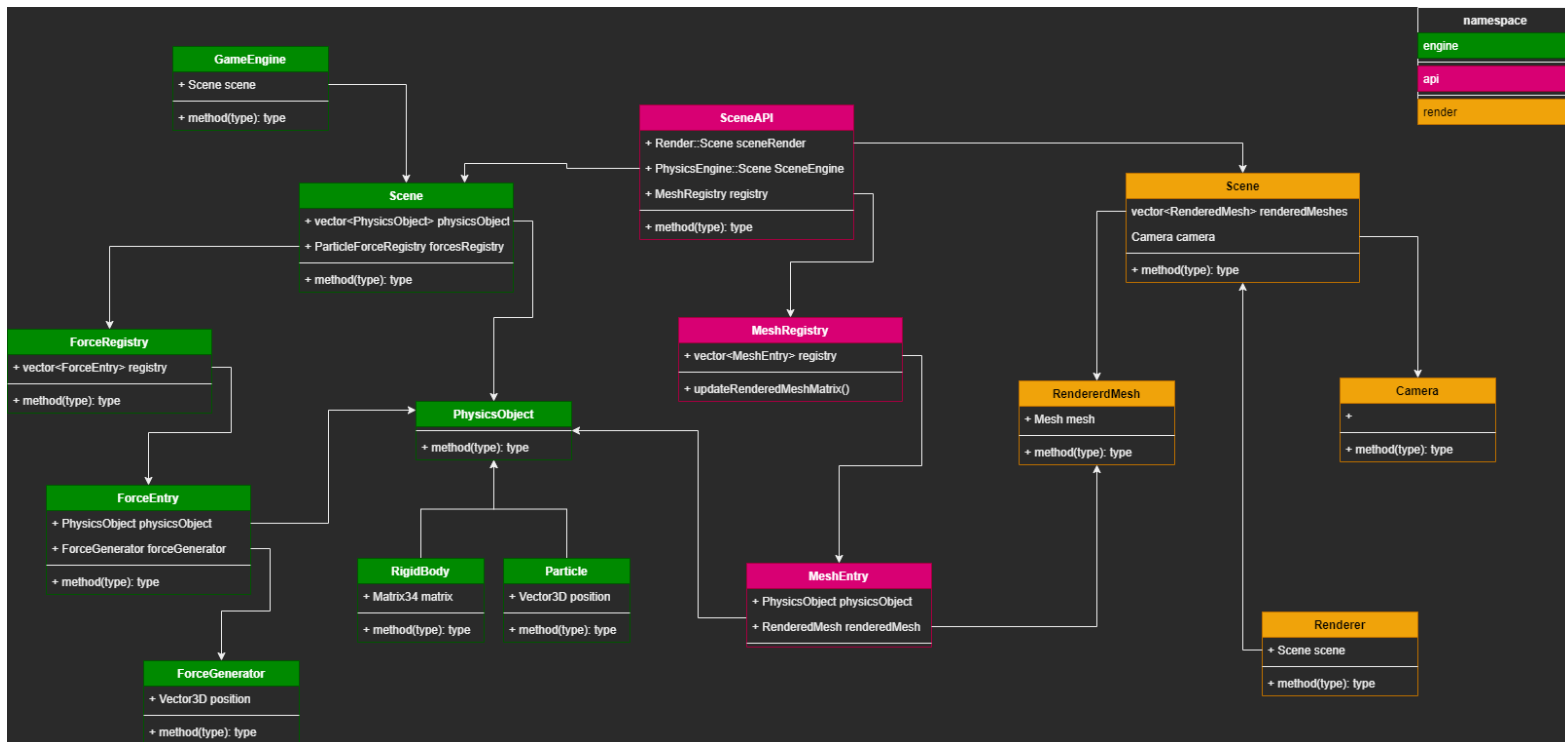
II/ Implémentation des forces

Après le cours sur les couples et moments d'inertie, nous avons tout le contenu de cours nécessaire pour faire l'intégralité du projet.

Nous avons également pu poser nos questions pour comprendre que la matrice de transformation n'était en fait qu'une transposition de l'état du corps rigide. Nous ne devons en fait pas l'utiliser pour modifier la position ou la rotation de notre objet physique. Pour l'objet que l'on veut afficher cependant, nous devons seulement

l'utiliser pour modifier la position et la rotation, il est toujours possible d'ajouter des transformations supplémentaires à l'objet que l'on rend.

Nous avons également pris le temps de nous mettre d'accord sur l'organisation générale du projet pour faciliter le lien entre le moteur physique et le moteur graphique. Voici donc l'organisation finale sur laquelle nous nous sommes mis d'accord :



Comme vous pouvez le voir, pour simplifier l'intégration de nos objets, qu'ils soient des particules ou des corps rigides, nous avons créé une classe interface entre nos deux types d'objets : PhysicsObject. Nous avons donc également créé une interface ForceGenerator pour les classes ParticleForceGenerator et RigidBodyForceGenerator. Ainsi, notre registry contient des entrées liant un ForceGenerator à un PhysicsObject et nous pouvons appliquer d'une même manière commune les forces sur nos PhysicsObjects.

Nous avons donc réparti les tâches :

- Célestin créera les interfaces communes aux particules et aux corps rigides
- Célestin et Eva continueront l'implémentation de la classe RigidBody avec les nouvelles méthodes que nous avons appris pendant le cours, notamment l'ajout des forces
- Célestin et Eva implémenteront les générateurs de force pour les corps rigides et adapteront les intégrations des PhysicsObject dans la scène avec les nouvelles interfaces
- Nicolas terminerait le lien entre le moteur physique et le moteur graphique

L'ajout des interfaces s'est très bien passé, et tout ce qui avait été fait jusque-là fonctionnait toujours bien.

Après quelques difficultés pour comprendre le passage des coordonnées locales aux coordonnées monde, Célestin et Eva ont fini par réussir à implémenter les générateurs de force.

III/ Mise en commun finale et départ sur les tests finaux

Lors du cours suivant, nous avons réussi à terminer le lien entre le moteur physique et le moteur graphique. Nous avons donc pu commencer nos tests de lancement de corps rigides et de "collisions" entre des corps rigides grâce au moteur graphique.

Nous avons au départ un bug qui faisait que les données des corps rigides évoluent bien mais ne mettaient pas à jour la matrice de transformation. Nous ne pouvions donc pas suivre l'évolution des corps rigides sur le moteur graphique. Nous avons rapidement corrigé ce bug en voyant que certaines lignes avaient été commentées en attendant la fin de l'implémentation des autres classes.

Nous avons donc commencé par lancer une voiture avec une vitesse initiale, et une vitesse angulaire initiale, et le tout fonctionnait relativement bien, mais avec certaines rotations précises (que nous n'avons pas réussi à identifier...), la voiture s'écrase et tourne instantanément à 180°. Nous avons passé des heures pour essayer d'identifier la provenance de ce bug mais nous n'avons malheureusement pas réussi à le corriger... La logique semble plutôt correcte et les rotations créées sont convenables mis à part ce "flip" des voitures dans certaines rotations...

Nous avons donc commencé à implémenter un autre programme de test pour simuler une collision entre deux voitures et les projeter en arrière avec une rotation. Pour ce faire, nous avons placé nos voitures en $x = -10$ et en $x = 10$, et nous avons ajouté une force de répulsion quand ces voitures arrivaient en $-2 < x < 2$.

Avec cette manière de faire, nous nous sommes rendu compte qu'une rotation était bien engendrée, mais les voitures n'étaient pas du tout propulsées en arrière. Après des heures de débogage, nous avons cru comprendre que ce problème était dû au fait que la force était ponctuelle, et était appliquée à notre accélération une seule fois, avec un Δt évidemment très faible puisqu'il correspond au temps entre deux frames (un ordre de 10^{-6}). Notre vitesse se retrouvait alors extrêmement peu changée. Nous avons donc décidé de garder les forces aux body points au moment de la "collision" puis de rajouter pendant un certain nombre de frames une force de recul au centre de masse de nos objets. Ainsi, nous avons une collision relativement réaliste. Pour un peu plus de réalisme encore, et pour bien présenter ce que nous avons fait, nous avons ajouté une force de gravité au moment de la collision pour faire tomber les voitures après qu'elles aient "sauté" légèrement.

Nous avons également essayé d'appliquer une force de ressort sur notre premier test, en lançant maintenant deux corps rigides avec une vitesse angulaire non nulle. Nous les avons donc liés par un ressort, et au moment de tester, nous nous sommes rendu compte qu'ils étaient bel et bien liés, et cherchaient à se rapprocher ou se repousser, mais leurs vitesses angulaires devenaient immenses à certains moments... Nous n'avons pas réussi à corriger ce problème... et nous avons donc décidé de rendre la force de ressort optionnelle et de laisser par défaut un simple lancé de deux corps rigides.