

Journal de Bord - équipe Mario

Lydie Santos, Roméo Louati, Evan Girard, Mathis Meunier

22 octobre 2024

Nous avons choisi de vous présenter ce journal sous forme de blocs dont chacun représente une séance de travail d'équipe. Chaque bloc détaille les actions réalisés ou commencé à cette date. Nous n'avons pas développé le travail individuel en dehors de ces sessions (finir notre partie du travail pour le rendez-vous suivant, réfléchir aux futures implémentations, ...).

Phase 1

11/09/2024

Où : Séance de cours

Temps passé : de 14h15 à 15h45

Résumé : Implémentation des classes Vector3d et Particle et affichage des particules à l'écran

Durant cette séance, nous avons démarré le projet en implémentant deux classes fondamentales : Vector3d et Particle. La classe Vector3d nous permet de représenter des vecteurs et des points dans un espace tridimensionnel, tandis que Particle encapsule les propriétés et comportements des particules que nous allons simuler. Pour chacune de ces classes, nous avons implémenté toutes les surcharges d'opérateurs nécessaires, ainsi que les méthodes attendues (produit scalaire, intégration, ...).

Nous avons également pris le temps d'établir des conventions de codage. Cela inclut le choix de coder en anglais, ce qui facilitera la compréhension du code pour tous les membres de l'équipe et d'éventuels contributeurs externes, et de commenter notre code selon les normes Doxygen. Cette approche nous permettra de générer facilement une documentation structurée et claire, si besoin est (cela facilitera ainsi la maintenance et l'évolution de notre projet à long terme).

13/09/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 17h

Résumé : Séparation des tâches de la phase 1 et séance d'implémentation en équipe

Durant cette séance, nous nous sommes répartis les tâches de la partie 1 restantes pour augmenter notre efficacité. Voici notre répartition :

- Lydie : vérification et complétion de la classe vecteurs + uniformisation du code + début d'implémentation des classes Matrix3 et Matrix4 (pour la suite du projet)
- Roméo : intégration + affichage des frames
- Evan : canon + gestion de l'affichage (gestion des axes, ...)
- Mathis : tests unitaires + projectiles

Durant cette séance, nous n'avons pas rencontré de problème majeur. Nous nous sommes mis dans une salle et avons continué le développement en échangeant lorsque nous rencontrions des problèmes.

18/09/2024

Où : Séance de cours + travail en groupe à la fin du cours

Temps passé : de 14h à 17h

Résumé : Correction des bug et début de fusion des codes

Nous avons consacré cette séance à finir nos tâches, selon les besoins, et à corriger nos bugs. C'est durant cette séance que nous avons rencontré notre première grande difficulté : la gestion des axes. Initialement, les axes sont en "mode ordinateur" (origine en haut à gauche avec l'axe Y vers le bas et X vers la droite). Nous souhaitions placer le repère en bas à gauche de l'écran avec l'axe Y vers le haut.

N'ayant pas réussi à réglé notre problème durant la séance, nous avons continué dans une salle. En croisant les informations, avec différents forums et la documentation d'open framework, nous avons réussi à atteindre notre objectif en utilisant diverses méthodes d'open framework pour déplacer le repère.

20/09/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 17h30

Résumé : Fusion des codes et récapitulatif

La première chose que nous avons faite lors de cette séance a été de vérifier l'uniformité de chaque code. Nous avons également passé en revue les tests afin de vérifier qu'il testait toutes les méthodes implémentées ainsi que leur cas limite.

Ensuite, nous avons fusionné la fin de nos codes (les différentes particules), avant de nous charger de l'affichage des frames (le dernier point à traiter pour compléter notre projet).

Enfin, nous avons complété ce journal de bord, qui retrace les difficultés que nous avons rencontrées ainsi que certains choix réalisés.

Globalement, la plus grosse difficulté de cette phase a été de s'approprier le fonctionnement d'OpenFrameworks, notamment le fait de gérer les repères ainsi que de créer notre HUD.

Il nous reste à finaliser les tests et à afficher les frames pour finir la première phase de notre projet.

24/09/2024 - Fin de la phase 1

Résumé : ReadMe + Rendu de la phase 1

Tous les derniers ajustements nécessaires étant finis et le ReadMe ayant été écrit, nous avons rendu notre projet sur Moodle. Lors de notre prochaine séance, nous discuterons de la marche à suivre pour mener à bien la phase 2 (plan de vue global du travail à faire, répartition des tâches, ...).

Phase 2

25/09/2024

Où : dans une collocation

Temps passé : de 14h30 à 16h

Résumé : Débrief de la phase 2 + répartition des tâches et ordre de priorité

Lors de cette séance de travail, nous avons commencé par faire un récapitulatif de la phase 2 et des attendus à réaliser. Nous avons donc commencé à nous répartir certaines tâches. Voici notre répartition :

- Lydie : registre des forces et ressorts
- Roméo : gestion des contacts et des collisions de type interpénétration
- Evan : gestion des collisions et des impulsions
- Mathis : force de gravité et forces de frictions

À la fin de la séance de travail, nous avons créé les premiers fichiers et commencé l'implémentation de la première force : la gravité.

04/10/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 17h

Résumé : Séance d'implémentation en équipe (et état d'avancement de chacun)

Durant cette séance, nous avons continué de développer nos parties de code. Nous avons pu finir le registre des forces ainsi que la force de gravité. Nous avons également pu implémenter les impulsions et la collision de type interpénétration.

La grande difficulté de cette séance a été de comprendre comment implémenter la collision. Nous nous sommes inspirés de la méthode d'implémentation des forces pour pouvoir implémenter les collisions (interface `particleContactGenerator`), puis nous avons créé différents types de collisions. Nous avons aussi commencé à réfléchir à une classe `particleContact`, qui gèrera les impulsions et le déplacement en cas d'interpénétration.

11/10/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h30 à 16h

Résumé : début d'implémentation des contacts et de la friction statique (Séance d'implémentation en équipe)

De même que pour la semaine précédente, la grande difficulté a été de voir comment nous pouvions implémenter la collision de type câble ainsi que les premiers contacts. Sachant que la

classe `particleContact` se veut assez modulaire, c'est-à-dire qu'elle fasse une impulsion peu importe le contact, nous avons eu du mal à passer à un contact de type "fil" (le câble dans notre cas).

Nous avons également pris le temps de discuter en groupe du principe d'interface, car il s'est avéré que tous les membres de l'équipe n'étaient pas familiers avec cette approche de codage.

18/10/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 18h

Résumé : Uniformisation + création d'une classe `World` + suite des forces

Nous avons commencé par examiner ensemble le concept de friction statique, car nous avons constaté que notre première approche ne tenait pas compte du fait que la normale pouvait être inclinée, entre autres. Pour remédier à cela, nous avons retravaillé cette partie du projet afin de garantir que les calculs soient justes, quel que soit le contexte de la friction statique. Enfin, nous avons créé une classe `Plane`, qui sera notre sol, et nous l'avons utilisée dans le contact au repos.

21/10/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h30 à 16h

Résumé : Mise en relation de toutes les parties du code + Début de préparation du rendu

Durant cette session, nous avons commencé la préparation de notre présentation. Nous avons, entre autres, commencé la réalisation des slides, en présentant nos avancées, les difficultés rencontrées tout au long de cette phase ainsi que leur résolution. Enfin, nous avons commencé à relier les différentes parties de notre code en préparation du rendu.

Nous avons remarqué que pour les ressorts, nous ne pouvons pas ajouter directement la force, il fallait indiquer une particule pour relier 2 particules avec un ressort. Nous nous sommes alors retrouvé avec un problème d'inclusion circulaire, que nous avons résolu en utilisant une "forward declaration".

Nous avons aussi fait la démonstration technique, un blob possédant des particules, pouvant se séparer et se reconstruire lorsqu'on appuie sur des boutons. Ceci sera spécifié dans le README.

22/10/2024 - Fin de la phase 2

Où : dans une colocation

Temps passé : de 15h à 17h30

Résumé : Rendu de la phase 2 + Entraînement de la présentation + Récapitulatif de la phase 2

Une dernière fois, nous avons vérifié l'uniformité de chaque morceau de code. Ensuite, nous avons vérifié que tout avait bien été réalisé et que notre build était fonctionnel. Enfin, nous avons relu le journal de bord afin de vérifier que tous les points importants du développement y étaient inscrits et nous nous sommes entraînés en vue de la présentation, avant de rendre notre projet.

Phase 3

31/10/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 13h à 16h

Résumé : Débrief de la phase 3 + répartition des tâches et début d'implémentation

Durant cette séance, nous avons commencé l'implémentation des classes suivantes : Quaternions et Matrix3/Matrix4. Ces dernières classes ayant déjà été commencées au cours de la phase 1, nous les avons simplement complétées.

Pour la classe Matrix4, nous avons fait le choix d'implémenter des matrices 4x4 au lieu des matrices 4x3 afin de permettre une autre utilisation de cette classe si nécessaire. Nous avons cependant adapté nos constructeurs pour qu'il nous soit possible de construire une matrice seulement en donnant les arguments d'une matrice 4x3 (la dernière ligne étant 0 0 0 1 par construction).

06/11/2024

Où : dans la salle de cours puis dans une salle à la bibliothèque de l'UQAC

Temps passé : de 13h à 18h

Résumé : Tests + Implémentation des corps rigides

Entre cette réunion et la précédente, nous nous sommes chargés de réaliser les tests demandés, soit l'ensemble des méthodes des classes Quaternions, Matrix3 et Matrix4. Nous avons donc commencé la réunion en vérifiant que tous les tests étaient bien complets et corrects.

Une fois les tests finis, nous nous sommes chargés de commencer l'implémentation de la classe rigidBody. En ce qui concerne la gestion des forces, nous avons repris le principe des forces appliquées aux particules que nous avons adapté à notre classe. C'est ici que nous avons commencé à rencontrer quelques difficultés.

Pour commencer, nous avons testé notre classe en appliquant uniquement une force de gravité. Or, celle-ci ne semblait pas s'appliquer à nos objets initialement. Nous avons donc cherché à trouver le problème et c'est ici que nous nous sommes rendu compte que nous avions un problème de référence. Finalement, nous avons réussi à appliquer la force de gravité à notre rigidBody. Il nous reste donc à gérer les forces appliquées à un point, ce qui entraîne une rotation de l'objet.

14/11/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 19h

Résumé : Forces appliqué à un point du corps rigide + Intégrateur physique

Durant cette réunion, nous avons géré les forces appliquées à un point du corps rigide. Dans ce cadre-là, nous avons ajouté une nouvelle force à notre moteur physique afin de pouvoir appliquer une force selon un vecteur prédéterminé à nos objets.

De plus, nous avons géré les effets de rotation de nos objets à l'aide d'un accumulateur de torque (qu'on a dû corriger, car notre première équation n'était pas correcte). Nous nous sommes également rendu compte qu'initialement, on testait uniquement si le point d'application était colinéaire à un axe de notre objet, or il faut que la force soit colinéaire également, car sinon la force appliquée doit bien faire tourner notre objet, ce qui n'était pas le cas avant avec notre implémentation.

En parallèle, nous nous sommes chargés de l'intégrateur physique afin de permettre la mise à jour de nos corps rigides. À la fin de la séance, nous avons fini l'implémentation générale du projet et il nous reste donc à créer le jeu de tir ainsi qu'à réaliser le support de présentation.

18/11/2024 - Fin de la phase 3

Où : dans une colocation

Temps passé : de 14h30 à 17h30

Résumé : Jeu de tir balistique + Support de présentation + Rendu phase 3

Premièrement, nous avons fini l'implémentation du jeu de tir balistique qui sera présenté lors de notre oral. Ensuite, nous avons tous relu le code afin d'uniformiser le code et de vérifier que les commentaires étaient bien écrits. Enfin, nous avons fini les slides de présentation et relu le journal de bord pour vérifier que toutes les difficultés rencontrées et les étapes de notre développement y étaient bien inscrites. Une fois satisfait de notre travail, nous avons rendu notre projet.

Phase 4

27/11/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 13h à 16h

Résumé : Débrief de la phase 4 + répartition des tâches et début d'implémentation

Durant cette séance, nous avons commencé l'implémentation de la classe Octree. Pour cela, nous avons créé une méthode qui gère la création de nouvelles feuilles ainsi que l'ajout des rigidBody à notre Octree, ainsi qu'une méthode qui gère les collisions entre rigidBody présents dans les feuilles et une méthode draw pour afficher notre octree.

4/12/2024

Où : dans la salle de cours puis dans une salle à la bibliothèque de l'UQAC puis dans une colocation

Temps passé : de 16h à 21h

Résumé : Fix de la structure d'Octree + Gestion des collisions

Premièrement, nous avons commencé la réunion en corrigeant notre structure d'Octree. En effet, nous avons remarqué qu'initialement, nous avons mal géré l'ajout des rigidBody dans l'Octree. En effet, ils s'ajoutaient tous dans le nœud principal et aucune feuille n'était créée. Nous avons donc repensé notre méthode pour corriger ce problème et, ainsi, créer des feuilles et transférer les rigidBody d'une feuille devenue nœud vers de nouvelles feuilles.

Nous avons également ajouté une nouvelle méthode à notre classe Octree qui nous permet d'actualiser celui-ci.

Entre cette réunion et la précédente, nous nous sommes chargés d'implémenter la classe RigidBodyContact, qui se base sur ParticleContact mais qui est adaptée aux rigidBody. Cela nous a donc permis, en fin de réunion, de commencer l'implémentation de la gestion des collisions. Il nous reste la gestion des rotations à implémenter.

Nous nous sommes également chargé d'implémenter un algorithme de résolution d'un programme linéaire (Fourrier Motzkin) pour déterminer s'il n'existe pas de plan séparateur entre deux objets. Nous nous sommes inspirés d'un code open source [disponible ici](#). On utilise ce code pour la détection de collision : on donne en entrée deux vecteurs de points : un à gauche, avec la contrainte que l'image par le plan est négative, et l'autre à droite, avec la contrainte opposée. Ainsi, si on trouve un résultat non valide ou nul, on sait qu'un plan séparateur n'existe pas.

6/12/2024

Où : dans une salle avec un tableau blanc à la bibliothèque de l'UQAC

Temps passé : de 14h à 17h

Résumé : Gestion des rotations lors de contact entre rigidBody

Comme nos contacts étaient déjà implémentés, il nous restait à implémenter la rotation. Pour cela, il nous a fallu adapter notre classe RigidBodyContact. Nous avons notamment eu besoin d'ajouter un attribut *contactPoint* pour définir le point de contact des rigidBody, et, par conséquent, appliquer la rotation correspondante.

10/12/2024 - Fin de la phase 4

Où : dans une colocation

Temps passé : de 15h à 18h

Résumé : Jeu de tir balistique + Support de présentation + Rendu phase 4

Premièrement, nous avons fini l'implémentation du jeu de tir balistique qui sera présenté lors de notre oral. Ensuite, nous avons tous relu le code afin d'uniformiser le code et de vérifier que les commentaires étaient bien écrits. Enfin, nous avons fini les slides de présentation et relu le journal de bord pour vérifier que toutes les difficultés rencontrées et les étapes de notre développement y étaient bien inscrites. Une fois satisfait de notre travail, nous avons rendu notre projet.