

Journal de bord

Session : Automne 2021

Présenté à : M. Hugo Trambly



Université du Québec
à Chicoutimi

Mathématique et physique des jeux
vidéo
8INF935

15/09/2021 première rencontre :

Mise en place du git, et de programmation de la classe vector3 chacun de son côté

22/09/2021 deuxième rencontre :

Mise en commun des classes vector3 et particle. Choix d'une librairie graphique (OpenGL)

27/09/2021 troisième rencontre :

Revue du code, de la partie graphique, mise en place du fps

28/09/2020 quatrième rencontre

Séparation inputs/logic/display, commentaires et écriture propre du code et fichier Readme et identification

06/10/2021

Préparation pour la phase 2 du projet :

Le git a été nettoyé. Nous avons supprimé les branches personnelles de chacun des membres et créées d'autres branches, une branche phase_1 et une phase_2 car cela avait plus de sens. Nous avons rendu l'arborescence du projet plus propre. Nous avons créé différents dossiers selon leur thématique (inputs, engine et graphics) avec à l'intérieur un dossier include et un dossier source. Cependant, nous n'avons pas eu le temps de renouveler les chemins de includes et réinclure les fichiers correctement dans le projet sur visual. Le projet n'est donc pour l'instant pas fonctionnel même la phase 1. Nous nous sommes mis d'accord sur des normes de code pour éviter les quiproquos et permettre une meilleure compréhension de celui-ci.

classe : MaClasse

variable:

private : m_nomVariable

protected : p_nomVariable

public : nomVariable

methodes : MaMethode()

struct : NomStruct

Enum : NomEnum

```
{  
    VALEUR_ENUM  
}
```

constantes : MA_CONSTANTE

Pour finir nous avons créé un projet sur git avec les différentes tâches à faire. Nous avons pour l'instant énoncé 8 tâches :

- Organiser le code dans des dossiers
- Editer le README
- Créer l'interface ForceGenerator
- Créer la classe ForceRegister
- Créer la classe ForceAccumulator
- Faire les 3 étapes de la boucle
 - o Appel des générateurs
 - o Calcul de forces
 - o Passage au moteur
- Mettre à jour l'affichage

07/10/2021

- Importation des fichiers dans le projet sur visual et changement des dossiers d'en-tête et chemin à l'intérieur du code de manière que la nouvelle arborescence soit prise en compte lors de la compilation. Le projet compile parfaitement. Fermeture de la tâche « organiser le code dans des dossiers »

23/10/2021

Claire a commencer à créer une classe d'accumulation des forces. Cependant après réflexion, on s'est rendu compte qu'il valait mieux l'intégrer dans la classe particule. On a donc ajouté à la classe particule l'attribut m_accumForces avec ces accesseurs et la méthode addForce qui vient juste additionner la force en paramètre aux force déjà accumulées.

24/10/2021

Implémentation des générateurs de forces et de l'enregistreur. Il n'y a pas eu de difficulté rencontrée au début. Création de la branche ForceGenerator. Nous avons commencé par générer la gravité et le drag puis les différents ressorts.

25/10/2021

Une fois l'implémentation fini nous avons intégré ces forces au système et avons eu différents problèmes. Dans un premier temps l'enregistreur de force pointait sur des coordonnées vides car le constructeur prenait un vecteur de 10 forces automatiquement. Nous l'avons corrigé. Une condition sur les forces jugées trop petites les supprimaient et du coup elles n'étaient pas calculées du frame suivant. Nous l'avons aussi enlevée. Ensuite nous avons tester le ressort ancré qui nous générait aussi une erreur liée à la normalisation des vecteurs qui ne prenait pas la condition d'une norme nulle.

25-26/10/2021

Implémentation de la gestion des collisions. La première difficulté a été de repérer comment procéder à la création de la classe de contact, voir à quoi correspondait chaque attribut... Une fois cela fait, un questionnement c'est fait quant au calcul de l'impulsion en fonction du poids des particules (ex : collision particule et terre, terre ne doit pas bouger). Un autre questionnement a été sur l'emplacement de la résolution des collisions. Faut-il le faire dans le résolveur ou dans le contact en lui-même. Nous avons choisi de le faire dans la classe contact car c'est elle qui connaît mieux le système ce qui rend les calculs et la compréhension plus simple. Pour finir, nous avons eu du mal dans la relecture du cours à retrouver les bons vecteurs normaliser et dans le bon sens. Nous n'avons pas eu de difficulté pour les autres classes. Pour le résolveur, nous avons voulu optimiser la boucle cependant par manque de temps et d'expérience, nous avons abandonné.

Une fois les classe de collisions implémenter, nous les avons intégrés au système. Rien ne fonctionnait. Le premier problème venait de la résolution de l'interpénétration lié à un problème de formule. Le deuxième problème, venait du fait que l'on déplaçait la particule trop loin. Pour régler cela nous a pris en compte la position de la particule ainsi que son déplacement. Le troisième, on s'est rendu compte que les impulsions se faisaient dans le même sens. Une fois ce problème résolu, nous avons essayer de créer un sol à l'aide d'une particule de la taille et de la masse de la Terre mais nous n'avons pas réussi à faire en sorte que le système de collision détecte un objet au repos.

10/11/2021

Commencement de la phase. Nous avons découpé la phase en différentes tâche puis les avons réparties.

16/11/2021

Première implémentation de matrice 3 et ajout des méthodes à la classe quaternion. La matrice3 a été longue car il fallait faire attention d'avoir les indices correctement. La vision d'implémentation des méthodes de la classe quaternion à été compliqué difficile à trouver car il a fallu comprendre correctement le comportement d'un quaternion.

17/11/2021

Nous avons vraiment débuté le projet et nous avons eu du mal à visualiser quoi faire. Les passages de référentiel du monde au local et du local au monde nous a demander du temps. A cela s'ajoute les outils à utiliser comme notamment la conversion avec quelle matrice utiliser et comment faire le calcul. Une fois que nous avons compris les passages de référentiel, nous avons commencé à implémenter la classe rigidbody.

22/11/2021

Implémentation de la matrice 4x4. Les difficultés étaient les même que pour la matrice 3x3. Le rigidbody n'a pas été fini car on a eu du mal à voir comment calculer les tenseurs d'inertie et même comprendre le principe de son utilisation. De plus, certaines parties de l'intégrateur n'ont pas été faite car il nous manquait des outils comme la matrice de transformation.

23/11/2021

Finalisation du projet. Nous avons fini l'implémentation de l'accumulateur de force et donc du rigidbody. Nous avons après mis à jour la logique et l'affichage pour faire les tests qui se sont avérer peut concluant. Nous avons quelques coquilles par si par-là dans différent outils mathématique notamment les matrices et les quaternions. Une fois les coquilles corrigées, nous nous somme pencher sur les différents calculs de l'intégrateur. Plusieurs soucis étaient liés à des problèmes de suivi de variable (pointeur, références, etc...). Ensuite, nous avons eu des soucis de passage de référence (local to world et world to local) car notre vision dans l'implémentation n'était pas correcte. Une fois ces soucis corriger, la phase a été terminée.

Pour conclure, les difficultés que nous avons rencontrées concernait la vision et la compréhension du problème. Cependant la similarité avec le 1^{er} tp on fait que nous avons moins de problème qu'à la phase 2.

01/12/2021

Merge de la phase 3 sur le main et mise en place des taches sur le git.

08-09/12/2021

Implémentation de la bounding sphere. Pas de soucis particuliers.

11/12/2021

Implémentation de l'octree. Il a été de compliqué de voir comment divisé les différentes zones d'un point de vue géométrique en grande partie lié à la 3D de la scène. Il a fallu schématiser plusieurs fois pour être bien sûr de l'implémentation et du découpage dans l'espace. La deuxième difficulté à été de savoir comment vérifier que la sphère était bien dans l'octree. Pas de problème en ce qui concerne la construction en arbre.

13/12/2021

Implémentation de toutes les classes concernant la narrow fase. Il a été compliqué de visualiser quelles primitives étaient engager dans le projet. Il été compliqué de détaché le concept de détection et de résolution par rapport aux données de collision à fournir. On a confondu collision et contact sachant que rien n'était encore relié. Lorsqu'on a fait la détection des points de contact, on a eu du mal à comprendre la démarche à adopter. On a encore eu des problème de géométrie à bien visualiser comment calculer le point de contact par rapport au plan.

14/12/2021

On créer la salle sans difficulté on juste mis du temps pour bien schématiser et adapter la boite à notre repère. Ensuite on a relié les différentes parties. On a récupéré la scène de la phase 3. On a ajouté au rigidbody la bounding sphere. Ça n'a pas créer de soucis

particuliers. Ensuite on a ajouté la détection de collision. Dans un premier temps, uniquement un plan sur les 6 était reconnue. On s'est rendu compte que l'ajout des plans dans l'octree ne se faisait pas correctement. Une fois cela régler un problème de géométrie a fait qu'uniquement la moitié des plans était détecter. La dernière difficulté concerne le découpage de l'octree. Il est trop petit et la collision est détecter 4 fois. On s'est rendu compte que ça ne concernait pas l'octree mais la bounding sphere qui était mal calculé et était trop grande.