

Projet : journal

Raphaël Vock

Lomàn Vezin

28 mai 2019

1 Progression

Tâches à effectuer

Temps imparti

✓ Création du journal	1 min
✓ Création du répertoire GitHub	5 min
✓ Lecture complète du descriptif général	1h
✓ Inscription en binôme	10 min
✓ Makefile	2h
✓ Classe Vector3D finie (pleinement opérationnelle, revisitée et testée)	4h
✓ Fichiers REPONSES et CONCEPTION	30 min / semaine
✓ Classe Particule finie (avec tous les ajouts)	5h
✓ Implémentation de Barnes-Hut	6h
✓ Premiers éléments	3h
✓ Plus d'éléments (mailles FODO : oui ; sextupôle : non)	2h
✓ Classe Accélérateur finie	3h
✓ Graphismes : Premiers pas	5h
✓ Graphismes : Particules	1h
✓ Graphismes : Éléments et accélérateur	4h
✓ Graphismes : Derniers ajouts	4h
✓ Classe Faisceaux finie (avec coordonnée curviligne)	4h
✓ Interactions inter-particule	4h
✓ Implémentation de Barnes-Hut	4h
✓ Résolution de beugues	4h

2 Suivi

Semaine 1

- Création du module Vector3D.
 - Testé et presque finalisé.
 - Mise en place du répertoire GitHub.
-

Semaine 2

- Mise en place de l'environnement Qt et réalisation du tutoriel graphisme P12.
 - Prise en main plus poussée de l'environnement graphique, implémentation de plusieurs exemples simples (polygones plans, cubes, sphères, sphères en mouvement).
 - Création du répertoire GitHub qui sera, à terme, le répertoire du projet final.
 - Résolution d'un bug très pénible du à qmake sur Macintosh qui rendait la compilation de librairie impossible (ouf!).
 - Création du module "physics" contenant un premier jet de la classe Particle.
 - Conception d'une toute première (et fort rudimentaire) simulation physique en temps réel à sortie graphique : un simulateur de problème à n corps (Calcul par somme directe, donc $\mathcal{O}(n^2)$. Devient très couteux à partir de 1,000 particules.).
 - Modification des opérateurs de calcul algébrique sur les vecteurs et surcharge des opérateurs d'affichage
 - Ajout de la méthode rotate à la classe Vector3D.
 - Ajout d'une méthode de la classe Particle permettant le calcul de forces électromagnétiques.
 - Création d'un fichier test pour la classe Particle, ce dernier écrit en sortie sur un fichier au format txt.
-

Semaine 3

- Cela sort un peu du cadre du projet mais je décide d'essayer d'implémenter l'algorithme de Barnes-Hut pour accélérer les calculs des interactions gravitationnelles des particules. Celui-ci est en $\mathcal{O}(n \log n)$ donc très intéressant si on a un grand nombre de particules en jeu.
 - Après avoir résolu un bug fort pénible, l'implémentation de Barnes-Hut est un succès. La simulation de problème à n corps pour 10^4 voire 10^5 particules peut maintenant être exécutée aisément même sur un ordinateur peu puissant. Évidemment l'algorithme peut facilement être adapté pour le calcul de forces électromagnétiques.
 - Finalisation de la classe Particle et écriture d'un fichier test.
 - Modification du fichier test des vecteurs, on préfère qu'il écrive en sortie sur une fenêtre terminal.
-

Semaine 4

- Première implémentation des éléments : une classe abstraite représente les éléments en général, de cette classe héritent deux sous classes pour les éléments droits et courbes.
 - Nous modifions par la suite notre conception pour limiter le nombre de classes.
 - Premier fichier test pour la classe accélérateur, erroné.
 - Correction du fichier test.
 - Débuts de la classe accélérateur.
-

Semaine 5

- Complétion de la classe élément et finalisation (temporaire) de la classe représentant l'accélérateur.
 - Premiers dessins de cylindres et sections de tore sur l'environnement Qt.
 - Ajout d'une classe pour la gestion des couleurs afin d'alléger le code.
-

Semaine 6

- Finalisation des dessins de cylindres et sections de tore, adaptation des méthodes au cas de notre projet.
 - Affichage des premiers éléments à l'aide des méthodes précédentes.
 - Révision de la conception des méthodes afin de rendre plus naturelle et pratique la construction graphique des éléments.
 - Ajout de namespace pour une gestion centralisée des exceptions ainsi que des constantes physiques.
-

Semaine 7

- Gestion des déplacements à la souris.
 - Ajout de la classe Faisceaux.
 - Révision complète des anciennes classes liées aux faisceaux.
 - Première simulation en mode texte.
-

Semaine 8

- Révision de la classe Particule, ajout d'une classe Point Charge pour rendre le code plus léger et intuitif.
 - Fin de l'implémentation des méthodes relatives aux faisceaux (émittance, coefficients des ellipses).
 - Ajout de la coordonnée curviligne de l'accélérateur (ainsi que la coordonnée propre à chaque élément).
 - Utilisation de cette dernière pour la l'implémentation des faisceaux circulaires distribués régulièrement.
-

Semaine 9

- Ajout de la cavité radiofréquence et des mailles FODO.
 - Réorganisation de la classe accélérateur.
 - Mise à jour de la simulation en mode texte.
-

Semaine 10

- Dernière révision de l’affichage graphique, nous optons pour un rendu plus épuré.
 - Ajout de différents points de vue afin de pouvoir mieux suivre l’évolution des particules (touches 1, 2, 3).
 - Révision complète de la gestion des événements clavier en conséquence.
-

Semaine 11

- Début de l’implémentation des interactions inter-particule à l’aide de l’algorithme de Barnes–Hut écrit principalement en semaine 3.
 - Test avec 7,500 particules, quelques ralentissements mais résultat très satisfaisant.
 - Ajout du mode matrice affichant les boîtes de l’algorithme relatives à Barnes–Hut (touche M).
-

Semaine 12

- Mise en page du fichier réponses sur \LaTeX , dessin des schémas représentant les différentes classes du projet.
 - Mise en page du fichier journal sur \LaTeX .
 - Relecture du code et ajout d’annotations, corrections mineures.
 - Révision des fichiers tests.
-

Semaine 13

- Résolution de quelques beugues et segfaults.
- Implémentation de distributions vectorielles aléatoires dans les faisceaux.
- Dernières finalisations, relecture et affinement de la documentation.