

# Projet: Journal

Raphaël Vock  
Lomàn Vezin

27 mai 2019

## 1 Progression

### Tâches à effectuer :

### Temps imparti :

✓ Création du journal	10min
✓ Création du répertoire GitHub	15min
✓ Lecture complète du descriptif général	1h
✓ Inscription en binôme	10min
✓ Makefile	2h
✓ Classe Vecteur finie (pleinement opérationnelle, revisitée et testée)	4h
✓ Fichiers REPONSES et CONCEPTION	15min/semaine
✓ Classe Particule finie (avec tous les ajouts)	5h
✓ Implémentation de Barnes-Hut	6h
✓ Premiers éléments	3h
✓ Plus d'éléments (mailles FODO, sextupoles, ...)	2h
✓ Classe Accélérateur finie	3h
✓ Graphismes : Premiers pas	5h
✓ Graphismes : Particules	1h
✓ Graphismes : Elements et accélérateur	4h
✓ Graphismes : Derniers ajouts	3h
✓ Classe Faisceaux finie (avec coordonnée curviligne)	4h
✓ Interractions interparticules (avec Barnes-Hut)	4h

## 2 Suivi :

### 2.1 Semaine 1 :

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

### 2.2 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 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 (Calculs par somme direct, donc  $\mathcal{O}(n^2)$ . Devient très couteux à partir de  $\sim 1000$  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 de la méthode force magnétique à la classe Particule.
  - Création d'un fichier test pour la classe Particule, ce dernier écrit en sortie sur un fichier au format txt.
- 

### 2.3 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.
-

## 2.4 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 des éléments afin d'éviter une duplication de code.
  - Premier fichier test pour la classe accélérateur, erroné.
  - Correction du fichier test.
  - Début de la conception de la classe accélérateur.
- 

## 2.5 Semaine 5 :

- Complétion de la classe élément et finalisation (temporaire) de la classe 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.
- 

## 2.6 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 meilleure gestion des exceptions ainsi que des constantes physiques.
- 

## 2.7 Semaine 7 :

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

## 2.8 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 conception des faisceaux circulaires.
-

## 2.9 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.
- 

## 2.10 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 évènements clavier en conséquence.
- 

## 2.11 Semaine 11 :

- Début de l’implémentation des interactions interparticulaires à l’aide de l’algorithme de Barnes-Hut implémenté en semaine 3 pour une meilleure complexité.
  - Test avec 7500 particules, quelques ralentissements mais résultat très satisfaisant.
  - Ajout du mode matrice affichant les boites de l’algorithme relatives à l’implémentation de Barnes-Hut (touche ‘m’).
- 

## 2.12 Semaine 12 :

- Mise en page du fichier Réponses sur  $\text{\LaTeX}$ , dessin des schémas représentant les différentes classes du projet.
  - Mise en page du fichier Journal sur  $\text{\LaTeX}$ .
  - Relecture du code et ajout d’annotations, corrections mineures.
  - Révision des fichiers tests.
- 

## 2.13 Semaine 13 :

- Dernières finalisations, relecture du projet dans son ensemble.
-