Journal de projet

Cappelletti William (MA) Machet Ludovico (PH)

— **Semaine 1** [22/02/2016-28/02/2016] :

Formation du binôme et prise de connaisance de la présentation du projet. Organisation des répertoires nécéssaires aux premiers pas du développement et création des fichiers texte demandés.

— **Semaine 2** [29/02/2016-06/03/2016] :

Réalisation d'une première version de la classe Vecteur3D, selon les indications données par les instructions de la semaine. La classe a été crée en suivant la modularisation proposée pendant le cours, avec une séparation nette des différents fichiers nécessaires, reliés enfin par un Makefile.
Premier contact avec l'environnement graphique : installation d' OpenGL et de Qt, lecture du tutoriel fourni jusqu'au premier exemple.

— **Semaine 3** [07/03/2016-13/03/2016] :

Vérification de la structure du Makefile déjà codé et petites révisions de la classe Vecteur3D, en particulier liées à la réecriture de certaines méthodes.
Continuation de la lecture du tutoriel graphique, jusqu'à l'exemple 4.

— **Semaine 4** [14/03/2016-20/03/2016] :

Profonde révision de la classe Vecteur3D et surcharge des différents opérateurs. Vérification du bon fonctionnement des surcharges et révision du fichier TestVecteur associé.
Écriture de la classe Ressort, selon les indications données, et du programme de test associé. Mise à jour du Makefile.
Conclusion de l'étude du tutoriel graphique.

— **Semaine 5** [21/03/2016-03/04/2016] :

Ècriture de la classe Masse et première compilation des différents test requis à ce stade du projet. Révision et amélioration des classe Masse et Ressort afin d'un fonctionnement optimal des tests. Revision globale de l'ensemble du code.

— **Semaine 6** [04/04/2016-10/04/2016] :

Mise en place et écriture des deux premiers intégrateurs et réalisation des tests associés. Premier contact avec gnuplot et réalisation du premier test en

suivant l'exemple fourni. Codage de la classe Tissu.

— **Semaine 7** [11/04/2016-17/04/2016] :

Révision et optimisation des classes des intégrateurs selon la phylosophie polymorphique et écriture du troisième intégrateur. Optimisation des classes précédentes pour un correct fonctionnement de la classe Tissu.

— **Semaine 8** [18/04/2016-24/04/2016] :

Codage des classe Dessinable et SupportADessin, selon les instructions fournies, et de la classe Systeme et TextViewer. Adaptation des classes Tissus et Intégrateurs pour le fonctionnement correcte des testTissus.

— **Semaine 9** [25/04/2016-01/05/2016] :

Mise en place de la partie graphique, avec l'adaptation des classes GLWidget et VueOpenGL aux dessinables contenus dans le projet, tels que systèmes, tissus, masses, et ressorts. Cela a impliqué la redéfinition du constructeur de la classeGLWidget, pour pouvoir construire un ou plusieurs systèmes à dessiner pendant la création d'un GLWidget. De plus, dans VueOpenGL il a fallu spécialiser les méthodes dessine() propres à chaque type d'objet dessinable. Au premier abord, les masses ont été dessinées commes des cubes colorés.

— **Semaine 10** [02/05/2016-08/05/2016] :

Codage des nouveaux tissus, en particulier définition complète des TissuChaine et TissuRectangle et prototypage des classes TissuDisque et TissuComposee. Implémentation de méthodes et codes utiles aux tests pour cette partie, comme la méthode void accroche(unsigned short int i = 2) dans TissuRectangle.
Du côté graphique, addition du dessin des axes du référentiel et codage de différents points de vue initiaux pour la caméra. De plus, représentation des masses avec des sphères pleines et implémentation de la possibilité de se déplacer dans la vue avec la souris. Le tout selon les conseils du tutoriel graphique.

— **Semaine 11** [09/05/2016-15/05/2016] :

Achèvement des tissus les plus avancés, codage d'une première version du TissuDisque et du TissuComposee. La classe TissuDisque a causé quelque diffuculté pour obtenir une correcte construction du tissu, qui apparaissait elliptique au contrôle graphique. Le problème a perduré pour toute la semaine. Parallèlement on a continué le travail sur la partie graphique, pour avoir des simulations plus jolies. En particulier, on a ajouté la possibilité de dessiner un tissu en dessinant et ses masses et ses ressorts, ou seulement un des deux. Cela se fait en appuyant sur la touche P, ce qui augmente un compteur dédiée dans la classe Systeme influençant la méthode void evolue(double const&) de la même classe.

— **Semaine 12** [16/05/2016-22/05/2016] :

Correction du constructeur de TissuDisque pour faire face à l'erreur discuté précédemment. Implémentation de l'intégrateur de Newmark et réalisation de test graphique pour en vérifier la stabilité, qui s'est révélée être assez précaire. Pour éviter une explosion immédiate de la simulation, il a été nécessaire de diminuer considérablement le pas de temps fourni par la classe GLWidget. Cela a permis d'obtenir de bons résultats même avec cet intégrateur. Codage de la classe Contrainte et de toutes ses sous-classes demandés par les consignes du projet. Conception de quelque extension liée aux contraintes. Ajout d'une fenêtre graphique (grâce à un QLabel) permettant l'affichage du nom de l'intégrateur utilisé couramment dans la simulation et ajout de l'événement clavier permettant de changer d'intégrateur.

— **Semaine 13** [23/05/2016-29/05/2016] :

Implémentation de l'intégrateur de Runge-Kutta, qui a rendu nécessaire l'ajout d'une méthode à la classe Masse, de prototype Vecteur3D force_totale, retournant la force subie par une masse avec position et vitesse données, utile pour calculer les accélérations intermédiaires pour cet intégrateur. Codage des extensions des contraintes, les Objets, qui intéragissent de façon plus réalistique avec les tissus, ce qui a suggéré de rendre les contraintes des masses, ce qui se traduit avec une relation d'héritage de la classe Contrainte de la classe Masse. Ces extensions ont causé des modifications assez importantes des classes Systeme, Tissu et Masse.