AIGREAULT Clément

BELLUOT Vincent

DUHAMEL Corentin

HENRIO Jordan

MONTOLIU Teddy

05/02/2013

Projet Génie Logiciel 2 : Espaces vectoriels, Applications linéaires et Polynômes

Sommaire

I – Introduction

Après le premier projet de génie logiciel du premier semestre et qui s'appuyait sur nos connaissances en bases de données, nous avons pour notre deuxième semestre un nouveau projet génie logiciel qui cette fois-ci s'appuiera sur les connaissances que nous acquerrons tout au long du semestre. Pour ce projet nous aurons besoin de plus de connaissance que pour le premier livrable car nous utiliserons ce que nous allons apprendre dans les cours de programmation objet, d'algorithmique et d'analyse orienté objet. Nous aurons aussi besoin de nos connaissances en mathématiques pour concevoir nos algorithmes.

Nous allons donc devoir développer un programme en java qui sera capable de gérer les espaces vectoriels et les polynômes, mais nous détaillerons plus en détail les capacités du programme dans ce livrable.

Pour ce premier livrable, nous devons dans un premier temps, reformuler de manière précise l'énoncé aussi bien sur le plan mathématiques que sur le plan informatique. Dans un deuxième temps nous donnerons une organisation de notre projet suivant la méthode de management : « Six O » et répartirons les tâches entre les membres du groupe à l'aide du logiciel Microsoft Project.

Nous allons maintenant commencer par la reformulation des besoins.

II – Reformulation des besoins

Pour le projet de ce semestre, nous devons développer un logiciel qui permet de faire des opérations sur des polynômes, des espaces vectoriels et des applications linéaires en relation avec ces notions.

Tout d'abord quelques définitions :

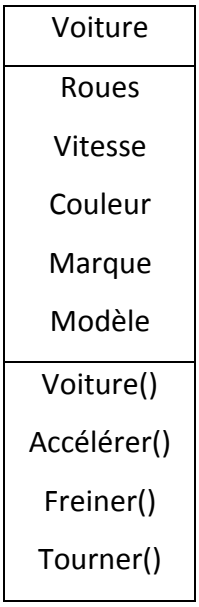
- Un espace vectoriel est un ensemble muni d'une structure (c'est-à-dire muni de plusieurs règles mathématiques) permettant d'effectuer des combinaisons linéaires (certaines opérations caractéristiques des espaces vectoriels).
- Un polynôme est une expression formée d'une combinaison linéaire de produits de puissances entières d'une ou de plusieurs indéterminées. Un polynôme s'écrit toujours de la forme suivante : $a_0+a_1X^1+a_2X^2+\ldots+a_nX^n$.
- Une application linéaire est une application entre deux espaces vectoriels qui préserve les combinaisons linéaires.

Chaque notion mathématique présentée ci-dessus présente des propriétés caractéristiques.

Notre logiciel comportera toutes les propriétés de chaque notion ainsi que leur définition, c'est-à-dire que notre programme devra dire si ce que demande l'utilisateur est bien définie ou non. Dans le dernier cas, il ne pourra pas faire les études qu'il désirera.

Pour ce qui est de la partie informatique pure, il nous est demandé d'utiliser le langage orienté objet, Java. Ce langage est très utilisé car il est compatible avec tous les systèmes d'exploitation (Linux, Mac, Windows, …) ainsi que de nombreux système du quotidien, comme les voitures, les micro-ondes, etc…

On parle de langage orienté objet lorsque, grossièrement, on peut représenter informatiquement des classes de cette manière :



Exemple d'une classe voiture

Ici notre classe, comporte un nom (Voiture), des paramètres (Roues, Vitesse, Couleur, Marque, Modèle) et des fonctions qui lui sont associés et qui modifierons son état (Voiture(), Accélérer(), Freiner(), Tourner()). Ensuite quand on créer une voiture dans un programme on parle d' « instancier un objet voiture ».

Il nous est aussi demandé de développer notre logiciel en suivant le pattern MVC (Modèle Vue Contrôleur). Tout d'abord, on parle de pattern, pour désigner une manière d'organiser son code. Dans le cas du pattern MVC, celui-ci consiste à séparer le modèle (qui représente les données), la vue (qui représente l'interface graphique, ce que l'utilisateur voit) et le contrôleur (qui représente la manipulation des données en fonction des interactions entre l'utilisateur et l'interface).

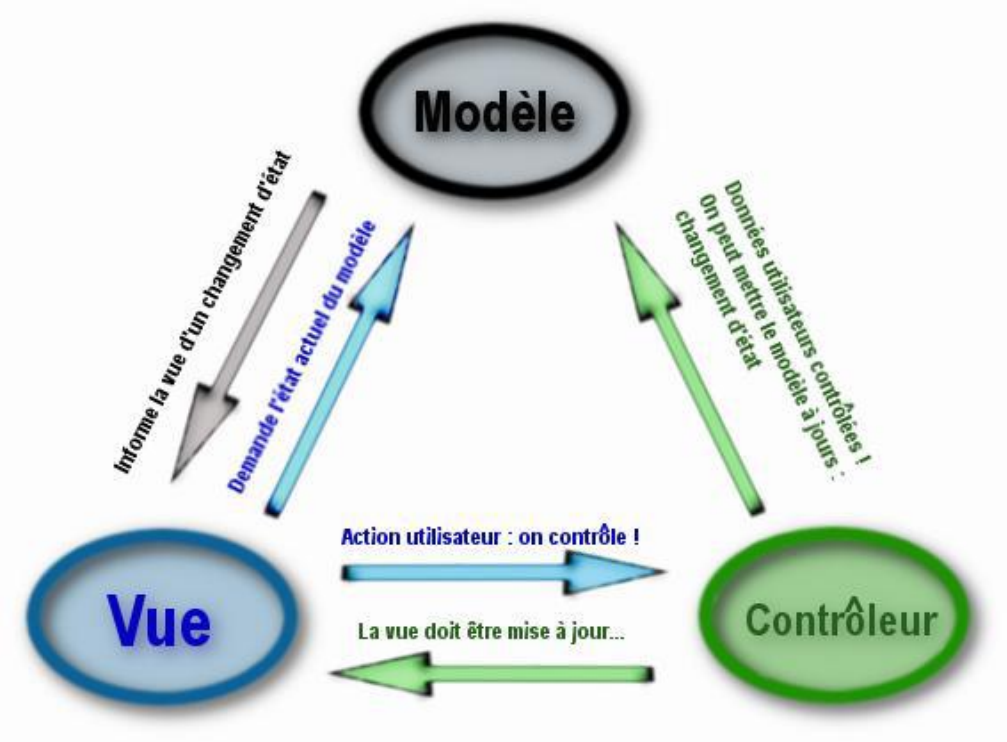


Schéma du pattern MVC

Et pour finir, nous devrons aussi mettre en place un système de sauvegarde des données pour que l'utilisateur puisse reprendre son travail lors de prochaine utilisation de l'application.

III – Organisation du projet

1. Application de la méthode « Six O »

Lors du séminaire d'organisation d'entreprise, nous avons abordé une notion d'organisation de projet nommée « SixO ». Le nom de cette méthode provient de la découpe qu'elle fait du projet : Objectifs, Objets, Opérations, Ordre, Opérateurs et Outils. Elle est employée, en général, en début de projet afin de mieux visualiser les besoins du client, déterminer les moyens dont on dispose et ceux dont on a besoin, planifier le déroulement du projet dans sa totalité.

Appliquons maintenant cette méthode à notre projet :

Quels sont les Objectifs à réaliser ?

Notre objectif est de réaliser une application qui manipule des espaces vectoriels, des applications linéaires et des polynômes. La réalisation de ce projet doit se faire par étape, nous nous donnons des objectifs avec un temps de conception. Ce projet, s'étendant sur une durée longue, ne peut pas se faire sans un minimum de hiérarchisation. Notre conception des objectifs est simple : obtenir un résultat convainquant avec une utilisation « agréable » pour un public extérieur. Nous mettrons en place tous les outils nécessaires à une conception originale, professionnelle et ergonomique.

Quels sont les Objets à mettre en place ?

Notre application doit se présenter à l'utilisateur par une **interface graphique**, facile à prendre en main sans pour autant être incomplète. Le logiciel doit être capable de **sauvegarder** les données introduites par l'utilisateur ainsi que les manipulations, pour que ce dernier puisse les retrouver à chaque utilisation du logiciel. Il doit bien sûr **manipuler les données** tout en respectant les règles définies par les mathématiques.

Quelles sont les Opérations à faire ?

Nous aurons besoins de faire plusieurs opérations afin de mettre en place ce logiciel. Nous devrons faire des recherches précises sur les sujets mathématiques concernés, pour pouvoir bien maitriser le sujet et aussi pour avoir une idée de la conception des algorithmes. Nous ferons aussi une première approche de notre application de manière schématique, à l'aide d'UML et d'algorithme en pseudo-code (algorithmes sur feuille). Et nous pourrons ensuite nous lancer dans le développement de l'application.

Dans quel Ordre les réaliser ?

1. Recherches mathématiques
    1.1. Espaces vectoriels
    1.2. Applications linéaires
    1.3. Polynômes

2. Première approche de l'application
    2.1. Représentation schématique
    2.2. Pseudo-code des méthodes de résolution

3. Développement de l'application
    3.1. Mise en place du contrôleur
    3.2. Mise en place de l'interface
    3.3. Mise en place du système de sauvegarde

Quels sont les Opérateurs qui interviennent ?

Nous serons les seuls opérateurs qui interviendrons sur le projet. Mais pour permettre une meilleure organisation entre les opérateurs nous utiliserons le gestionnaire de projet Git. Certains membres du groupe ont déjà une expérience avec Git, et il s'est avéré très utile. C'est pour cela que nous comptons l'utiliser une fois de plus.

Quels sont les Outils dont nous avons besoin ?

Nous aurons besoin de Git, comme expliqué précédemment, ainsi que du JDK de Java, qui permet de créer des logiciels dans ce langage.

Voici, d'après notre application de la méthode SixO, comment se déroulera notre projet.

Nous avons utilisé Microsoft Office Project afin de développer un planning complet des tâches que chaque opérateur devra effectuer.

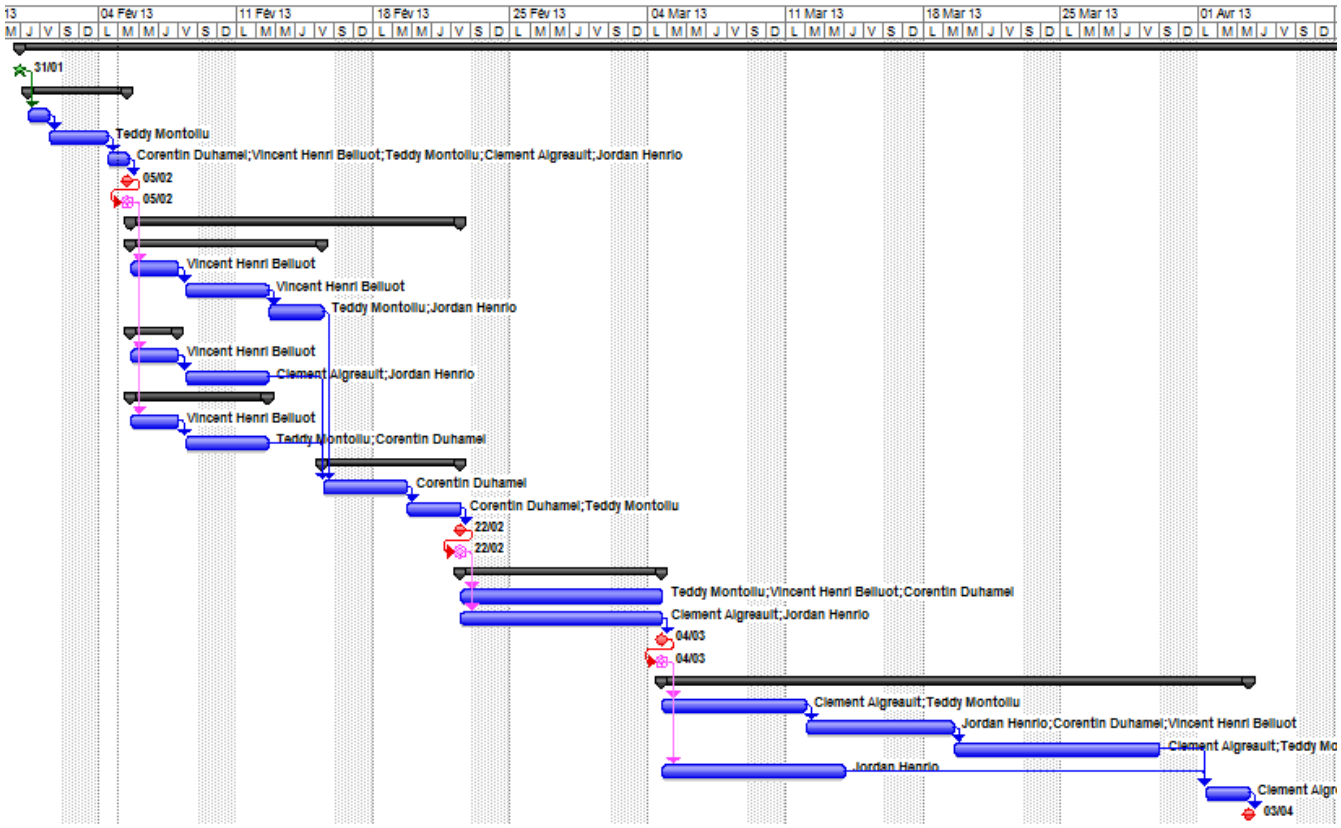
2. Application sous Microsoft Project

		Code	Objets et opérations	Ordre	Durée
3		**1.2**	**Livrable 1: Cahier des Charges**		**8 jours**
4		1.2.A	Methode SixO	2	3 jours
5		1.2.B	Elaboration du planing du projet	4	2 jours
6		1.2.C	Redaction du Cahier des charges	5	3 jours
7		Jalon	Rendu livrable 1	6	0 jour
8		Jalon	Debut livrable 2	7	0 jour
9		**1.3**	**Livrable 2 Analyse**		**30 jours**
10		**1.3.1**	**Etude des Espaces Vectoriels**		**18 jours**
11		1.3.1.A	Recuperation des données necessaires	8	6 jours
12		1.3.1.B	Elaboration de la classification des Espaces Vectoriel	11	6 jours
13		1.3.1.C	Affectation des attributs des espaces	12	6 jours
14		**1.3.2**	**Etude des applications lineaires**		**6 jours**
15		1.3.2.A	Recuperation des données	8	6 jours
16		1.3.2.B	Affectation des attributs des applications lineaires	15	6 jours
17		**1.3.3**	**Etude des Polynomes Formels**		**12 jours**
18		1.3.3.A	Recuperation des données	8	6 jours
19		1.3.3.B	Affectation des attributs des polynômes formels	18	6 jours
20		**1.3.4**	**Etudes des methodes de calcul**		**12 jours**
21		1.3.4.A	Recuperation des données	19;16;13	6 jours
22		1.3.4.B	Elaboration des compatibilitées	21	6 jours
23		Jalon	Rendu livrable 2	22	0 jour
24		Jalon	Debut livrable 3	23	0 jour
25		**1.4**	**Livrable 3: Travail préliminaire**		**15 jours**
26		1.4.1	Ecriture des algorithmes en pseudo-code	24	15 jours
27		1.4.2	UML	24	15 jours
28		Jalon	Livraison du livrable 3	27	0 jour
29		Jalon	Début Livrable 4	28	0 jour
30		**1.5**	**Livrable 4 : Conception**		**51 jours**
31		1.5.1	Serialisation des objets en JAVA	29	12 jours
32		1.5.2	Codage des Algorithmes de calcul	31	13 jours
33		1.5.3	Système de sauvegarde	32	20 jours
34		1.5.4	Création de l'interface	29	17 jours
35		1.5.5	Test et debug	33;34	6 jours
36		Jalon	Rendu livrable 4	35	0 jour
38		**1.6**	**Livrable 5: Finalisation**		**8 jours**
39		1.6.1	Rédiger séparément les remarques individuelles	37	4 jours
40		1.6.2	Faire une synthèse collective des remarques	39	4 jours
41		Jalon	Rendu livrable 5	40	0 jour
42		Jalon	Fin du projet	41	0 jour

Prévision du planning

Nous avons divisé le projet en 5 livrables. Le premier livrable, regroupe la méthode SixO, l'élaboration du planning et l'introduction au sujet. Le deuxième regroupe le travail d'analyse mathématique, c'est-à-dire, l'étude des espaces vectoriels, des applications linéaires et des

polynômes formels, ainsi que l'application des méthodes de calcul. Le troisième livrable nous prépare au quatrième. En effet, nous ferons l'UML et pour faciliter la programmation en JAVA, traduire notre analyse en algorithme pseudo-code. Le livrable 4 sera consacré à la conception JAVA, préparé préalablement. Et dans le livrable 5, nous rédigerons nos rapports personnels et collectifs.



Emplois du temps

Nous avons donc pour objectif le rendu des 5 livrables dans les temps. Pour des raisons personnelles de plusieurs membres du groupes nous avons ajouté quelques jours de plus pour certaine tâches. Nous avons choisi de répartir le travail en fonction des compétences de chaque membre du groupe. Nous avons aussi prévu un délai de 20 jours avant la date finale de rendu, pour être sûr de pouvoir parer au mieux les imprévus.

IV – Conclusion

Nous avons donc, comme annoncé dans l'introduction, reformulé le sujet ainsi que présenté notre organisation pour la réalisation de ce dernier. Toutefois il est possible que cette organisation change en fonction d'une éventuelle avance ou, le cas échéant, un éventuel retard.

Par la suite nous construirons le projet étape par étape, sûrement en quatre ou cinq livrables. Nous pensons faire, dans les livrables à venir, des recherches mathématiques concernant les sujets que l'on devra être capable de manipuler nous présenterons donc ces recherches. Nous ferons une première approche de notre logiciel en UML. C'est-à-dire que nous représenterons schématiquement le fonctionnement de notre application. Et nous nous attaquerons ensuite au développement de cette dernière.

Ce rapport se trouve sur notre Git, le « tuteur » de notre groupe y sera invité, pour pouvoir suivre l'avancement du projet.