Projet SI3 : 2010 - 2011

Gestionnaire de tournois sportifs

Équipe treize
Sylvestre Genevier
Antoine Pultier
Roman Mkrtchian
Stéphane Muller
Dimanche 30 janvier 2011

Table des matières

Présentation de l'équipe, et répartition des tâches

Sylvestre Genevier

Après avoir vu quelques notions d'héritage, Sylvestre s'est chargé de coder les classes de base assez simples, à partir d'une architecture définie par toute l'équipe.

Antoine Pultier

Antoine est chargé de la sauvegarde des objets de base du programme en XML, et en base de données Sqlite.

Roman Mkrtchian

Roman a pour tâche de créer l'interface homme machine en console.

Stéphane Muller

Stéphane est le chef de projet. Il a aussi pour tâche de créer les fonctionnalités du projet en s'appuyant sur les class codées par Sylvestre.

Présentation

Le projet consiste à réaliser un programme de gestion de tournois sportifs. Il se base sur un projet similaire réalisé auparavant.

Cette fois-ci, il faut ajouter des fonctionnalités au programme, et changer certains aspects de modélisation. De plus, les équipes ont totalement changées. Au lieu de binômes, il y a désormais quatre membres par équipe, et personne n'a travaillé de la même façon.

Déroulement et organisation

Dans notre équipe, il a été décidé de recommencer un projet de base, en s'inspirant fortement de nos anciens codes sources respectifs. Il aurait été possible de partir d'un code source d'une personne, mais au moment de commencer, nous n'avions pas les retours des enseignants correcteurs de l'ancien projet.

Il était donc impossible de savoir qui avait l'architecture la plus propre. Nous avons donc décidé de créer une nouvelle architecture en tenant compte de nos expériences acquises précédemment, et des remarques lancés par certains enseignants lors des oraux de présentation de l'ancien projet.

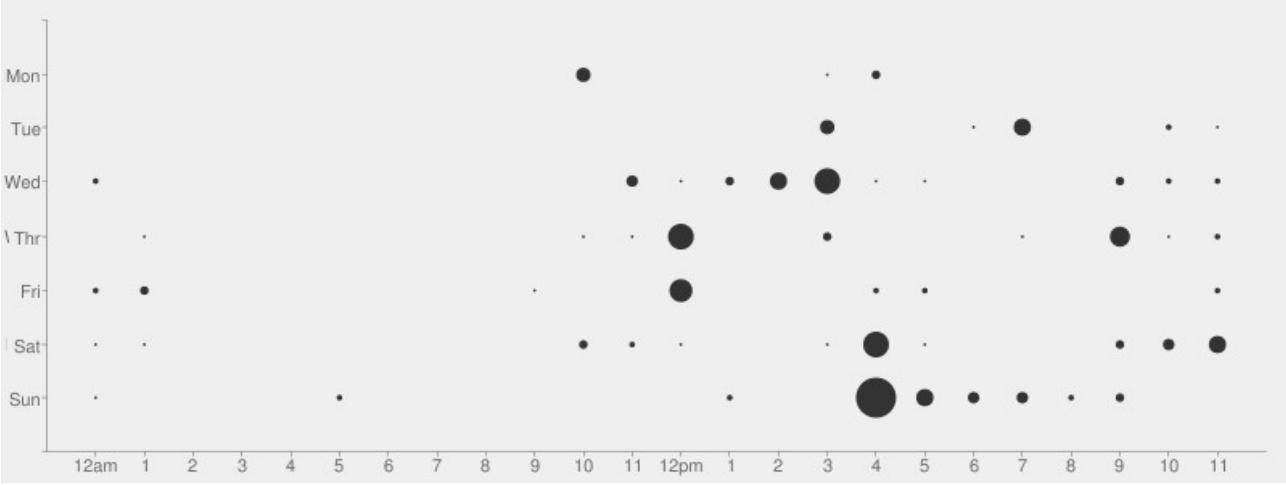
La nouvelle architecture a été créée le premier jour, en équipe. L'idée était que le code de base puisse être créé le plus vite possible.

Une fois l'architecture déterminée, les tâches ont étés réparties en fonction de la motivation de chaque membre.

Un chef de projet a été élu par la même occasion, il s'agit de Stéphane Muller.

Pour des raisons pratiques, l'équipe a décidé de travailler à domicile, mais une réunion a été programmé à heure fixe tout les jours de la semaine, au site des Templiers.

Le travail a été réalisé relativement uniformément au cours de la semaine, comme en témoigne ce graphique tenant compte du code publié en fonction du jour de la semaine, et de l'heure. C'est jolie, mais pas très lisible, je vous l'accorde.

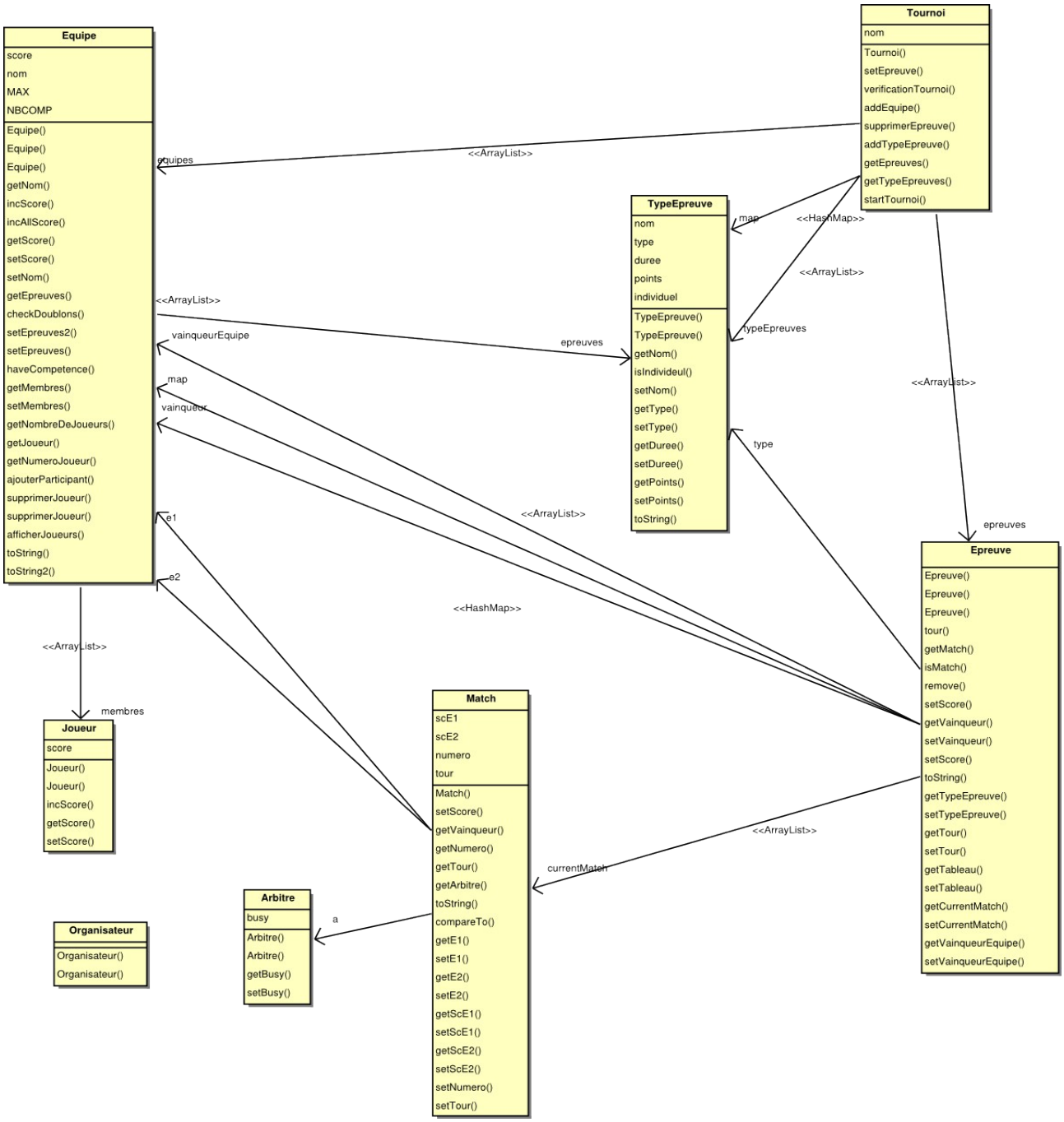


Architecture du projet

L'architecture du projet est plus compréhensible avec des schémas. Après le déni de certains enseignants de l'école pour la méthode Merise, il est présenté des schémas UML...

Schéma global de l'application

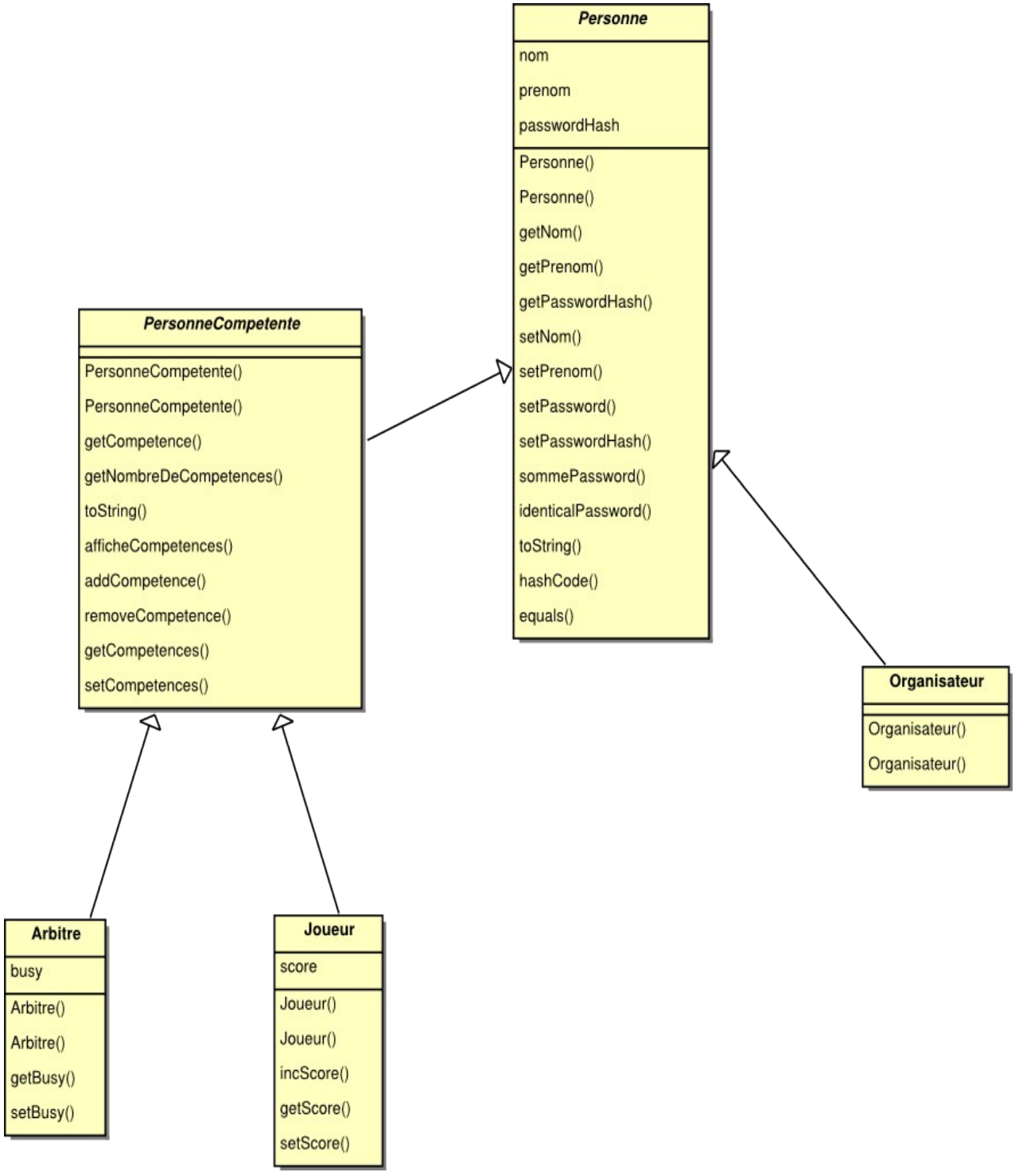
L'architecture globale est relativement simple : des équipes avec des joueurs, des tournois avec des équipes et des épreuves, des épreuves avec des matchs, et des matchs avec deux équipes et un arbitre. S'ajoute à cela l'organisateur qui est l'administrateur de l'application, et la gestion des différents types d'épreuves.



Il faut signaler que l'équipe a accordé une grande importance au sens du détail, et que les réunions ont permis de faire de longues discussions sur certains aspects de la conception de l'architecture.

Architecture des personnes

Cette architecture est d'un niveau technique plus simple, et a été réalisée pour Sylvestre, débutant en Java. Cela lui a permis d'apprendre l'héritage, dans un cas concret et pratique.



Architecture du système de sauvegarde

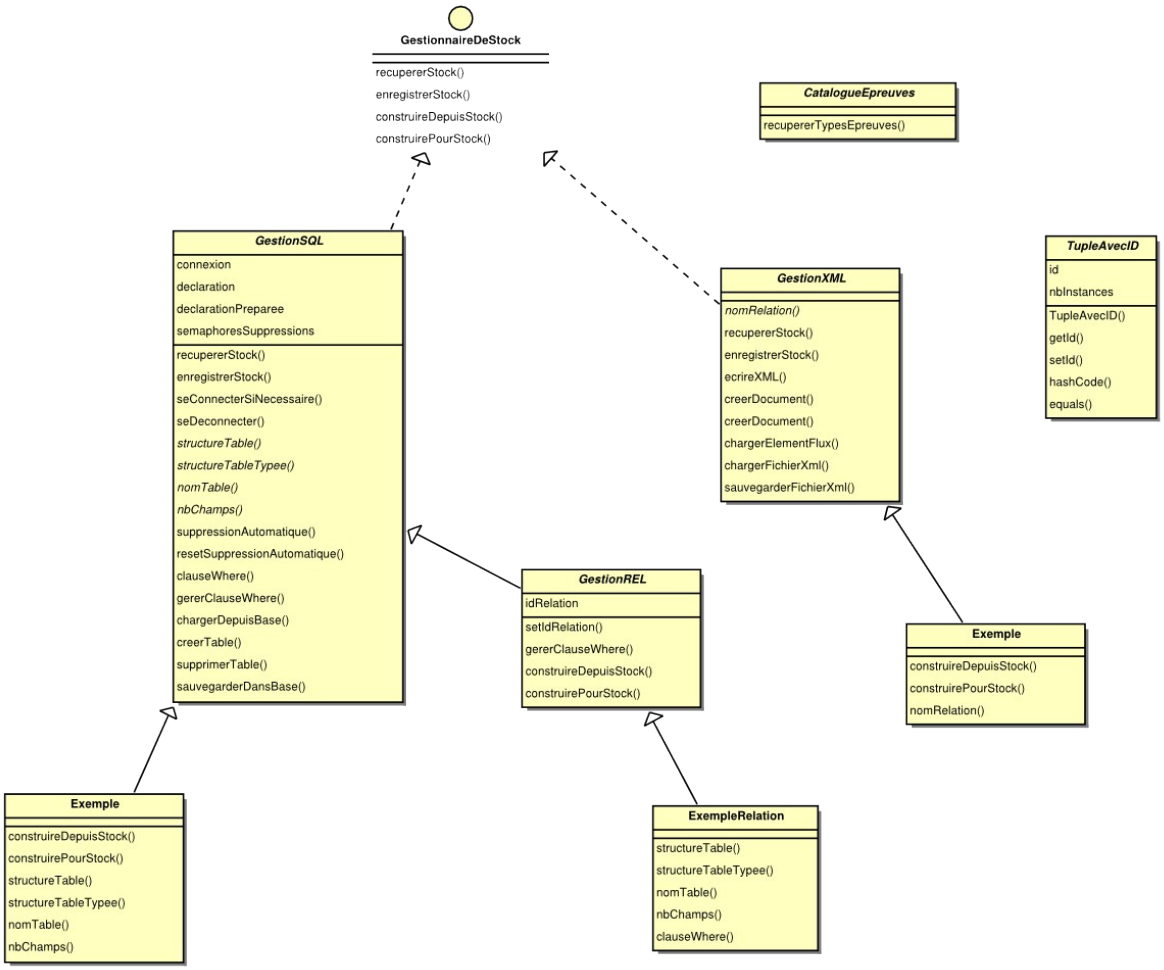
Le système de sauvegarde s'appuie sur une interface de gestion de stock commune à toute les méthodes de sauvegarde.

La sauvegarde et la création des objets est séparée dans des classes au même nom, mais précédé de la méthode de sauvegarde. On a par exemple JoueurSQL, et JoueurXML.

Pour regrouper le code et les classes de gestion de même type, il existe des classes abstraites GestionSQL, et GestionXML. GestionREL est une variante permettant de gérer les relations dans la base de données.

Pour pouvoir être stockés, les objets doivent hériter de la classe abstraite TupleAvecID.

CatalogueEpreuves sert à récupérer la liste des différents types d'épreuves, qui est stocké dans un tableau au format CSV.



Travail réalisé

Sylvestre Genevier

Sylvestre a réalisé comme prévu les différentes classes représentants des personnes, et les classes équipe. Il a eu de l'aide pour certaines nouvelles notions comme l'héritage, et quelques erreurs.

Il a aussi codé un programme dérivé qui construit un environnement de fonctionnement d'exemple.

Antoine Pultier

Antoine a réalisé l'architecture du système de sauvegarde, l'écriture du système de sauvegarde pour le XML, et le SQL de la base de données Sqlite. Il s'est aussi occupé du tableau de description des différents types d'épreuves, en utilisant le format CSV, et la bibliothèque libre OpenCSV.

Une fois ce travail terminé, il s'est occupé de quelques corrections de code, et de la réalisation de ce rapport.

Roman Mkrtchian

Roman s'est occupé tout au long du projet de l'interface console. Il a aussi participé activement aux débats sur les spécifications de l'architecture.

Stéphane Muller

Stéphane s'est occupé du développement des classes Épreuve, Match, Moteur (routines de fonctionnement), et Tournoi.

Il a corrigé d'autres classes, et s'est occupé du bon fonctionnement global du programme.

Pour cela, il a créé les principaux tests de fonctionnement utilisant la librairie Junit.

Il est aussi le chef de projet, il a envoyé de nombreux mails, et a passé beaucoup de temps à lier certaines parties du programme entre elles.

Gestion des Tests :

Sylvestre étant débutant, il ne connaissait pas junit.

Peter Sander, enseignant de l'école nous a confirmé que Junit n'était pas adapté aux tests sur les bases de données, et qu'en conséquence, il n'était pas nécessaire de tester la base de données et le XML.

L'interface console n'a pas lieu d'être testée non plus.

Conclusion

Maintenant qu'il est quasiment terminé, nous pouvons dire que le projet a été enrichissant en expérience. C'est un travail de groupe, sur une courte période, avec un cahier des charges assez flou.

D'un point de vue technique, le projet n'est pas compliqué. Mis à part Sylvestre qui a appris l'héritage, il n'y a pas d'apprentissage de nouvelles notions, mais juste l'application de méthodes de programmation, dans une nouvelle architecture. Cela s'explique surtout par le fait que le projet est très similaire à un précédent travail, et que la personne en charge des sauvegardes avait déjà travaillé dans ce domaine.

Cependant, passer du temps à plusieurs sur un tel programme est toujours riche en expérience.

On a aussi pu remarquer a quel point la différence de niveau au sein d'un groupe peut affecter le déroulement du projet.

Le point positif du déroulement du projet a été les réunions journalières qui nous ont permis de réfléchir ensemble à la façon dont le projet devait évolué.

Nous savons désormais que ce qui a été réalisé sur ce projet devra être réutilisé dans un autre projet, certainement similaire, mais assez différent pour devoir y passe beaucoup de temps aussi.

En conclusion, ce projet a été enrichissant, en attendant le prochain projet.