

Bilan

Etant une équipe de 6 personnes, nous nous sommes logiquement séparés en deux groupes afin d'effectuer les premières tâches, donc avec une sous équipe s'occupant de la partie Bases de Données, et l'autre de l'architecture logicielle. Nous avons essayé de communiquer au mieux afin de ne pas perdre le fil du projet. Chaque sous-équipe a réussi à bien comprendre l'enjeu du projet, et après quelques ratés, nous avons enfin une architecture logicielle satisfaisante, ainsi que le schéma relationnel. Vient le moment de commencer la programmation des deux exécutables, ainsi que la mise en place de la base de données.

La base de données étant la base du projet, nous avons décidé de placer deux personnes très rapidement dessus. Le raisonnement était avant tout de créer les différentes tables, les initialiser et vérifier que tout se passait bien. Nous avons décidé de ne pas mettre en priorité le développement de la connexion. Ainsi nous nous connectons et déconnectons à chaque requête. Nous pensions qu'il était nécessaire d'avoir une base de données fonctionnelle avant d'optimiser la connexion.

En parallèle, la deuxième personne s'est occupée de la vérification des déplacements des pièces. Le débogage fut très long car les vérifications sont très complexes, et nous n'étions pas au fait de certaines subtilités dans les requêtes. Cela nous empêchait de regarder au bon endroit. Cependant, nous sommes arrivés à vérifier les déplacements de toutes les pièces. Notre approche n'a toutefois pas été très rigoureuse. En effet, nous avons effectué les tests dans la *main* de la classe au lieu de créer des classes de test. Ceci s'explique en partie par le fait que nous avons développé les classes en reprenant le modèle donné sur Chamilo, et nous voulions effectuer des petits tests très rapidement. Pour de futurs projets, nous aurons le réflexe de réaliser systématiquement des fichiers de test. Nous nous sommes ensuite rapidement occupés de faire les requêtes nécessaires à la bonne gestion du tournoi, qui sont passées aux tests sans soucis.

Le problème majeur résidait dans l'avancée des autres tâches. Ces dernières ont mis beaucoup trop de temps à se lancer. En prenant en compte les différentes erreurs lors de la programmation et le temps de débogage, nous avons dû intervenir auprès des autres groupes 3 jours avant le rendu. Ainsi, nous avons dû travailler dans la précipitation. En couplant cela au fait qu'il fallait adapter les classes pour les utiliser dans l'interface graphique et les contrôleurs, nous n'avons pas pu tirer la pleine quintessence des requêtes SQL afin d'avancer rapidement au niveau de l'interface graphique et textuelle. Ce problème intervient notamment à cause du fait que la tâche E : conception objet et implémentation du coeur de l'application a commencé avec deux semaines de retard. Si la tâche E s'était finie à temps, nous aurions pu avancer sur l'interface textuelle, sans prétendre la finir avec succès étant donné la courte période disponible pour l'implémenter.

Pour ce qui est de l'implémentation logicielle, nous avons un squelette de code, et nous nous attaquons à l'interface graphique. Reprenant au départ une interface graphique utilisée

d'un ancien projet de POO, nous nous rendons compte que cela ne sera pas suffisant, il faut changer d'implémentation pour l'interface graphique. Nous sommes encore dans les temps et nous décidons d'utiliser Swing, il faut donc un temps d'apprentissage. L'une de nous se charge de maîtriser cet outil pour pouvoir travailler sur l'interface graphique. Deux autres personnes sont affectés à la base de données, et les 3 autres à l'écriture des classes principales.

A partir d'un moment, nous avons fait l'erreur de moins communiquer, et donc le travail s'en est trouvé ralenti. Des différences de vision du projet ont entraîné des erreurs de compréhension des tâches. Nous avons pris du retard sur le planning prévu, et nous avons essayé de rattraper ça en nous réunissant plusieurs fois sur de longues périodes. C'est vers la fin du planning que nous avons compris comment réellement mettre en commun nos parties, entraînant beaucoup de bugs. Heureusement, nous avons réussi à nous retrouver pour terminer correctement le projet, mais avec de grosses refontes sur le code.

Pour conclure, nous avons appris lors de ce projet que même si les tâches semblaient totalement distinctes, il faut toujours communiquer et avancer conjointement. Il est difficile d'avancer sur un tâche sans être au courant des tâches annexes, et cela peut porter préjudice. De plus, des idées peuvent être émises par d'autres membres du projet auxquelles on n'a pas pensé, et inversement. Malgré une fin difficile, ce projet nous a quand même apporté niveau prise en main d'une base de données, et maîtrise de l'architecture logicielle, avec la découverte d'une véritable interface graphique.