Synthèse personnelle

GL28 Dorian Houdelette

Janvier 2020

Introduction Début janvier j'étais assez content de pouvoir faire ce projet de groupe étant donné que les cours commencent de plus en plus à m'ennuyer. Et sincèrement j'ai appris beaucoup de choses. Étonnamment je n'ai fait que très peu de *Java* mais j'ai écrit pas mal de tests en *Deca* et les scripts correspondant pour automatiser la chose. J'ai également beaucoup travaillé sur l'extension et donc les méthodes de calcul des fonctions mathématiques usuelles, même si le manque de temps s'est fait ressentir sur le rendu final. Enfin, comme tout le reste du groupe, j'ai beaucoup appris d'un point de vue *Gestion de projet* i.e. la communication, le développement simultanée, le report de bug, ...

1 Travailler en mode projet

Je pense que notre gestion de projet a été très mauvaise au début. En effet, les cours étant très dispersés et le polycopié conséquent, nous étions vite perdus et ne savions pas vraiment quoi faire. Ainsi les premiers jours, chacun travaillait dans son coin sans être sur de ce qu'il faisait.

C'est seulement en fin de première semaine que nous nous sommes répartis les rôles de manière plus efficace et que nous avons mis en place plusieurs outils pour communiquer et faire part de notre avancement à tout moment. Nous avons en effet crée un *Trello* (cf. Suivi I) mais malheureusement personne ne s'en servait et donc je ne reviendrai pas dessus. Cependant nous avons crée un serveur *Discord* (qui n'est autre qu'un service de messagerie objectivement très bien agencé), qui nous a beaucoup aidé à communiquer en dehors des cours (lorsque la partie C devait poser une question par rapport à la partie B par exemple), à reporter des bugs (c'était une bonne chose de laisser une trace écrite des bugs pour éviter qu'ils tombent dans l'oubli), ou plus simplement pour préparer les suivis et soutenances.

Un point sur lequel je suis très fier de notre gestion de projet est notre adaptabilité. En effet, lors de la dernière semaine de projet alors que l'extension n'avançait pas, moi qui était chargé des tests et Locif qui était chargé de la partie A et qui ne savait plus quoi faire avons changé de rôles : je suis parti sur l'extension et Locif a pris en charge les tests car il se sentait incapable de faire des maths pour notre extension. Et c'est pour ça qu'en semaine 3, j'ai pu laisser Locif complètement seul sur les tests et me concentrer sur l'extension.

Cependant un point qui m'agace un peu plus est justement l'extension, en effet je me suis rendu compte que celle-ci était très en retard seulement quelques jours avant le rendu et j'ai du mettre les bouchées doubles pour rendre quelque chose de convenable

mais dont je suis très peu fier. En effet il y a pleins de concepts que j'ai à peine effleurer voir complètement ignorer et c'est fort dommage.

2 Validation

Une chose est sûre, ce projet m'aura appris à mettre en place des scripts automatisés, notamment en bash. Que ce soit pour tracer les courbes de l'extension (erreur et nombre de cycles internes) ou alors pour lancer notre base de tests.

En effet j'ai, en début de projet, passé pas mal de temps à mettre au points des scripts de tests qui fonctionnaient mais étaient en somme peu satisfaisant. Par exemple pour valider un programme de partie A ou B, j'imposais qu'on mette en commentaire selon un certain format l'arbre ou alors les tokens attendu(s) en sortie. Mais ce format nous a causé beaucoup de problèmes car un simple espace manquant ou une simple erreur de location (lorsque l'on tapait à la main) pouvait faire rater notre test alors qu'il était bon. On a donc, à cause de ça, passé beaucoup de temps à chercher des erreurs là où il n'y en avait pas et c'est ainsi que j'ai décidé de modfier les critères de validation (cf. Documentation de Validation). C'est ainsi que je me suis rendu compte, que vouloir à tout prix tout automatiser n'était pas forcément un gain de temps et que parfois avoir une personne qui passe un peu plus de temps à regarder la sortie de chaque programme (au moins une fois) était, bien que plus laborieux, beaucoup plus efficace.

En fin de projet, je me suis chargé de l'extension et j'ai ainsi été contraint de comparer les fonctions que nous avons écrite à une valeur de référence. Nous avons choisi de prendre les fonctions de Java.Math en guise de référence. Un problème se posait à ce niveau car nous ne savions pas à quel point ces fonctions étaient précises (même si on supposait qu'elles l'étaient car Java est quand même un langage très reconnu dans le milieu professionnel). Et c'est ainsi que nous avons trouvé que concernant Java, pour les fonctions en flottant, elles sont toute précises à un ULP près. Ainsi pour valider nos fonctions nous n'avions plus qu'à tracer l'erreur en ULP et décider si cela était suffisant ou non pour nous. De même j'ai également écrit un script qui permettait d'écrire un programme *deca* qui calculait une fonction en une valeur donnée, l'executer et à l'aide de ima -d, obtenir le nombre de cycle interne nécessaire au calcul. Ainsi pour l'extension, nous avions deux critères de validité : coût et précision (cf. Documentation de l'Extension).

Conclusion Ces deux semaines ont donc été très bénéfiques autant d'un point de vue *technique* que d'un point de vue *relation client*. Cependant il est dommage que nous ayons atteint un rythme efficace d'avancement seulement au bout de deux semaines de projet. En somme, je suis assez fier du travail qu'on a produit. Certes notre compilateur possède un nombre non négligeables de défauts, mais je suis certain qu'avec quelques jours de plus, nous aurions pu produire un compilateur bien meilleur.