



ÉCOLE NATIONALE DE L'INFORMATIQUE ET DES MATHÉMATIQUES APPLIQUÉES DE GRENOBLE

Projet Génie Logiciel

Bilan individuel de compétences
Année universitaire 2020-2021

Le projet Génie Logiciel a été l'occasion de progresser sur de nombreux points. Il faut avant tout rappeler dans un premier temps que c'est le projet le plus conséquent que j'ai pu produire, avec un investissement important et beaucoup d'attentes. En sachant cela, ce projet est déjà constructif puisqu'il m'a forgé sur des points que je me contenterais de citer, comme la gestion du stress, l'organisation personnelle ou celle en équipe sur un projet aussi conséquent, me donnant ainsi un aperçu intéressant et complet de ce que pourront être des projets professionnels à l'avenir. Dans ce document, je vais développer notamment deux compétences que j'ai acquies ou renforcé, en justifiant par des exemples concrets rencontrés lors de ce projet. J'ai choisi de détailler ma capacité à mettre en oeuvre des processus de validation qui a été primordiale dans la conception de ce compilateur, et j'illustrerais les exemples et tâches qui m'ont permis de comprendre les implications des calculs faits par une machine.

Tout d'abord, comme tous les membres de l'équipe j'ai participé activement à l'élaboration de la base de test, et ce pas seulement pour la partie sur laquelle je travaillais activement (à savoir la partie de génération de code assembleur [Partie C]). En ce sens, j'ai aiguisé ma capacité à établir un ensemble de test complet, en essayant pour chaque test d'isoler au mieux un point du fonctionnement à valider, ce qui n'est pas toujours facile (par exemple pour des instructions plus avancées comme les boucle while ou les blocs conditionnels, il faut être certain que des étapes plus basiques (déclaration de variable, opérations booléennes et arithmétiques) ont été correctement implémentées, et donc testées). De plus, l'opportunité offerte par ce projet avec l'outil jacoco m'a permis de voir les failles de base de test avec les pourcentages de couverture et les détails de ces derniers. Avoir ces données m'a permis de m'améliorer par moi même et de déceler seul mes oublis, ce qui aide l'assimilation et la progression.

J'ai aussi beaucoup progressé quant à l'automatisation des tests. Ici, le travail en équipe a été très bénéfique, j'ai pu tirer l'avantage de camarade déjà expérimenté pour les scripts de test, mais j'ai aussi complété leur aide par des recherches personnelles. Au final, j'ai rapidement pu automatiser mes tests et gagner en efficacité en les incluant à l'outil maven. J'ai aussi pu être témoin de l'intérêt de faire des tests réutilisables au cours de la conception. Entre autre, tous nos tests de la partie sans objet du langage étaient toujours inclus dans la base de test pour le langage complet, et ils ont servis de bases pour des tests sur la partie objet (Par exemple, les tests de boucle while et de bloc conditionnel (sans objet) ont été adaptés pour être testés dans des méthodes de classe (objet)).

La partie des tests est centrale dans le processus de validation, mais en tant qu'élément moteur de la partie C j'ai aussi beaucoup utilisé les outils de débogage, que ce soit celui de la machine virtuelle IMA (option **-d**) ou bien celui d'Eclipse. Ces outils se sont avérés être d'une aide précieuse et j'ai assimilé leur importance pour mener à bien un projet. En effet, à de nombreuses reprises nous avons pu régler des problèmes grâce à cet outil. Il s'agissait souvent de problème de mémoire, l'exécution pas à pas permettait de trouver rapidement l'instruction à la source du problème.

Enfin, travailler sur la dernière étape d'un projet où les étapes dépendent des précédentes m'a formé pour gérer les contraintes de temps notamment. En effet, c'était souvent cette étape qui

était attendue, et en décalée par rapport aux deux premières. J'étais donc l'un des principaux acteurs de notre capacité à valider notre avancement du projet en respectant les contraintes de temps.

Pour finir, si je devais refaire ce projet en voulant améliorer ma progression pour cette compétence, j'apprendrais encore plus en détails le langage bash pour faire des scripts de tests plus performant, spécialement au sujet de la comparaison avec les fichiers résultats (nos scripts comparaient les résultats d'exécution avec des fichiers de références), qui dans le cadre de ce projet était très rigide, alors que des options de la commande **diff** me permettraient probablement de faire mieux.

Ce projet, et plus précisément mon travail sur la partie C m'a offert la chance de comprendre finement les implications des calculs faits par une machine.

Premièrement, j'ai grandement progressé au niveau de ma compréhension et de mon utilisation des langages orientés objet. En effet, j'ai majoritairement codé en langage java, ce qui a accru mes compétences pour ce langage, mais surtout, le compilateur compile du langage deca, qui est aussi un langage à objets. En ce sens, j'ai revu les notions centrales d'héritage, polymorphisme, liaison dynamique. L'architecture de notre projet était bien sur une grande opportunité de se plonger dans des familles de classes complexes, avec de l'héritage omniprésent, et donc des réécritures de méthode ou du polymorphisme en permanence (cf doc Conception, méthodes codeExp qui est un exemple parmi beaucoup d'autres). Tous les tests, bien que simples étaient aussi l'occasion d'écrire en deca et donc de manipuler un langage orienté objet.

Une des compétences techniques que j'ai le plus renforcé grâce à ce projet a été ma compréhension de la mémoire d'une machine, au travers des notions de pile, tas, pointeurs de piles (GB, LB, SP). Notamment, lorsque j'ai dû concevoir la partie génération de code pour notre extension (TAB), les tableaux avaient une adresse sur la pile pointant un bloc sur le tas, avec à la première adresse de ce bloc une adresse vers un second bloc. Ainsi, j'ai beaucoup affiné cette maîtrise, et j'aurais beaucoup aimé la pousser encore plus loin en m'intéressant à la désallocation des tableaux (et objet ?).

Un point sur lequel j'ai passé beaucoup de temps, et que j'ai donc beaucoup travaillé, fût mon assimilation du langage assembleur de la machine virtuelle. J'ai désormais une compréhension très nette de la notion de registre, des branchements ou encore comme je le mentionnais précédemment, des constituants de la mémoire d'une machine. La gestion des étiquettes en assembleur lors de l'étape de génération de code a été très formatrice, et on voit vite à quel point les quelques instructions de nos langages de programmation habituels deviennent des blocs conséquents d'instructions d'assembleurs, d'où aussi l'importance de l'optimisation, point aussi que j'aurais aimé plus approfondir, bien que j'en ai déjà saisi l'importance.

Enfin, j'ai pu constater les limitations du calcul sur les flottants. Déjà, ces derniers sont beaucoup plus sujets à des débordements et des instabilités numériques (notamment pour la division), et dans le cadre de notre compilateur, les flottants génèrent beaucoup plus d'instructions (pour vérifier les débordements arithmétiques) que les autres types de variables (entiers ou booléens). De plus, j'ai aussi pu travailler en fin de projet sur les bibliothèques de calcul matriciel, et ici

cette instabilité et ces limites sont plus que palpables. Dans le cadre d'algorithmes classiques d'algèbre linéaire, comme le calcul d'inverse ou bien la recherche de valeurs/vecteurs propres les erreurs d'arrondis sont déjà visibles lorsque nous faisons des démonstrations simples (un ou deux calculs successifs). Ainsi il est clair que j'ai désormais conscience des limites à considérer pour des calculs plus importants impliquant les flottants.

Pour conclure, ce projet Génie Logiciel a été riche d'enseignements et je suis fier de ce que j'ai pu apprendre et du résultat fournis par mon équipe. J'ai la sensation d'avoir appris d'un côté par moi-même, et de l'autre grâce aux compétences diverses de mes camarades. J'ai développé deux axes importants, surtout liés à des champs techniques, mais j'ai aussi évolué humainement dans ma communication, et mon organisation au sein d'une équipe de 5 personnes.