

Bilan individuel de compétences

Romain Pierre

Janvier 2020

Durant le projet GL j'ai eu l'occasion de découvrir de nouveaux aspects de la gestion de projet, de retrouver des automatismes acquis lors du projet C et d'améliorer mes connaissances sur la compilation d'un programme. Afin de dresser un bilan plus précis, je vais réaliser une introspection sur deux compétences auxquelles j'ai été particulièrement sensibilisé.

1 Travailler en mode projet

Durant près de 4 semaines, nous avons travaillé en équipe de 5 afin d'obtenir le compilateur le plus abouti possible. Néanmoins une organisation précise et clairement définie s'est imposée afin de mener à ce projet.

En effet, la première semaine a démontré le danger d'un manque d'objectifs fixés, sans quoi il fut difficile de briser l'inertie initiale. Établir des cibles ambitieuses dès le début du projet aurait beaucoup apporté à l'avancement du projet. De plus, communiquer ses attentes est crucial. Nous avons 3 différentes sources d'informations à l'origine : un Trello, un diagramme Gantt et un serveur Discord. Cela avait pour effet de disperser l'information puisque peu de membres du groupe ne consultaient le Trello, et le diagramme était assez vague donc peu utilisable en soi. La décision de tout regrouper sur Discord fut judicieuse car intuitive d'utilisation et adaptée à la communication instantanée.

Cela c'est notamment révélé crucial car nous n'arrivions pas à trouver un accord sur nos horaires de travail en présentiel à l'Ensimag, afin de faire des compte rendu sur l'avancement et les tâches à venir. Nous avons donc transposé ces compte rendus sur le Discord du projet, en profitant des fonctionnalités de marquage (mots en gras, souligné ou barré) pour suivre notre évolution.

Enfin j'ai remarqué qu'il est facile de perdre de vue certains points du projet lorsque l'on est focalisé sur une fonctionnalité à implémenté. Afin de prémunir cela d'arriver aux rendus et suivis, j'affichais régulièrement sur un channel à part les deadlines à venir ainsi que des captures d'écran du polycopié associées à ces deadlines. Cela permettait d'avoir plus rapidement accès à ces informations ainsi que de servir de rappels.

2 Comprendre finement les implications des calculs faits par une machine

A partir de la deuxième semaine, j'ai assumé le rôle de 'dev exclusif' de l'étape B. J'ai eu l'occasion de développer des connaissances solides des mécanismes d'héritage nécessaires à la gestion de l'arbre généré par la syntaxe abstraite et son débogage. En effet, l'utilisation des classes abstraites afin de gagner en généralité s'est révélée intéressante au cours du développement lorsque j'ajoutais de nouveaux terminaux. Je devais seulement implémenter les spécificités de ces classes, sans me soucier du reste du code qui restait correct.

De plus, notre approche en Test Driven Development m'a particulièrement sensibilisé aux erreurs renvoyés par mon code ainsi que l'utilité d'un message d'erreur explicite. Il en va de même l'affichage *prettyPrint* qui est déterminant pour vérifier l'exactitude de l'arbre résultant.

Enfin j'ai acquis une connaissance fine du processus de vérification contextuelle et de la grammaire attribué du langage Deca, qui reste globalement valable dans la plupart des langages orienté objet.