

# Bilan Individuel

GACEL Etienne

January 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Mettre en oeuvre des processus de validation</b>	<b>2</b>
<b>3</b>	<b>Agir en professionnel responsable</b>	<b>3</b>
<b>4</b>	<b>Conclusion</b>	<b>3</b>

## 1 Introduction

A travers ce document, je vais revenir sur ce que le projet GL m'a apporté d'un point de vue compétences.

## 2 Mettre en oeuvre des processus de validation

Les tests ont été un pilier fondamental dans notre réalisation du projet. Très rapidement, le nombre de tests à lancer rendait la validation de notre travail couteuse en temps. J'ai donc créé des scripts bash de lancement de tests automatisés. J'ai voulu ces programmes rapides d'utilisation et spécialisés.

Le premier challenge rencontré a été le langage bash. Avant le projet, j'avais peu de connaissance et surtout aucune expérience de programmation en bash (philosophie du langage, ses limites, sa difficulté à prendre en main, débuggage etc). Mes premiers scripts ont donc été simples (ce sont les scripts Lexical.sh, Syntax.sh et Context.sh). Ces trois premiers programmes sont inspirés des programmes bash fourni dans le projet. Ils m'ont apporté une première prise en main du langage, une vision des erreurs "bêtes" mais communes et une approche de debugging. J'ai ensuite approfondi mes nouvelles compétences en créant le programme Compilateur.sh, programme qui compile puis exécute tous les tests codegen puis qui compare la sortie de l'exécution avec un le contenu d'un fichier réponse crée préalablement par les membres de l'équipe.

Le deuxième challenge a été de trouver une méthode de validation des tests efficace (rapide et la plus automatisable possible). En fonction des scripts et des fonctionnalités testés, j'ai adopté plusieurs solutions pour répondre à cette problématique. Pour les tests de vérifications contextuelles, j'ai choisi de repérer si les test\_\* renvoient une erreur ou non ; la solution est simple. Par contre pour les tests de compilation, la solution est moins évidente. il est facile de vérifier si le .ass a bien été généré ou non mais lorsque l'exécution affiche un résultat, comment vérifier sa justesse ? La solution que j'ai apporté est de créer un fichier d'extension .ans de même nom que le fichier .deca testé contenant la sortie attendue du programme deca. Le script va donc comparer la sortie de l'exécution du .ass avec le .ans. Cette solution fonctionne mais est trop contraignante pour les utilisateurs : obligation de créer un .ans sinon aucun test ne se lance, obligation de porter le même nom que le .deca testé. A de nombreuses reprises, les membres du groupe ont oublié des .ans ce qui a diminué la fréquence d'utilisation du script et son intérêt. Cette problématique m'a donc permis de constater que la validation de tests n'est pas nécessairement évidente et qu'il est important que l'automatisation demande un minimum d'actions de la part des utilisateurs.

En conclusion sur ce point, toute la réflexion que j'ai dû fournir pour amener une validation automatique de notre compilateur m'a permis de développer d'une part mes connaissances et compétences en bash en plus de satisfaire à une cer-

taine curiosité mais aussi de constater que les méthodes de validation ne sont pas nécessairement évidentes.

### 3 Agir en professionnel responsable

Dans ce projet, j'ai travaillé principalement sur l'étape B du compilateur. J'étais en équipe avec Mohamed Ali LAGHA. J'ai été le *leader* de ce duo et je veillais un bon avancement de notre partie. J'ai très vite remarqué que Mohamed Ali était bon pour traduire les règles de grammaire contextuelle en méthode Java mais était plus lent en *debugging* et avait une moins bonne vision de tous les tests possibles. J'ai alors défini rapidement nos rôles. Mohamed Ali allait s'occuper de l'implémentation des vérifications contextuelles pendant que j'allais créer les tests correspondants et déboguer si les tests ne passaient pas. Lorsque Mohamed Ali me disait qu'il se sentait fatigué je l'ai conseillé de prendre du temps de repos. Quand il a dû rentrer en France et donc allait être indisponible pendant un certain temps, on s'est organisé ensemble pour palier ce temps perdu. Au final, en plus de gérer mes capacités et ressources, j'ai travaillé étroitement avec un collègue que je ne connaissais pas, analyser ses forces et faiblesses et composer un duo efficace et complémentaire.

### 4 Conclusion

Pour conclure, ce projet m'a permis de monter grandement en compétences. D'une part en compétences techniques ; mon niveau en Java s'est perfectionné et j'ai appris le langage bash de manière à avoir de premières bases. D'autre part au niveau managérial où j'ai pu pleinement gérer un duo.