Grenoble INP ENSIMAG

GL53

BILAN

Notre équipe:

Aboueloula Ayman

Benchekroun Omar

Bensrhier Nabil

Jedoui kacem

Yagouti Redouane

I. Organisation de l'équipe

L'organisation au sein de l'équipe et la répartition des tâches a été inspirée des compétences techniques de chacuns et des affinités par rapport aux parties du projet, on a choisi d'aborder la partie A ensemble pour entamer le projet sur les même bases et essayer d'avoir une vue globale sur le travail demandé et essayer d'équilibrer chaque partie pour qu'elle soit faite à temps, la communication et l'esprit de groupe ont constitué les atouts de notre équipe, tout en étant autonome et en respectant les détails de la charte d'équipe qu'on a établi en début de projet.

La conclusion qu'on peut tirer au niveau de la gestion et la répartition des tâches est qu'on est satisfait du travail fourni par chacun et le sérieux de l'équipe dans les phases critiques du projet, bien sûr notre compilateur n'est pas parfait et il reste des bugs qu'on a pas su gérer, mais sur le strict nécessaire qu'on a abordé avec la prof sur le dernier suivi on est plutôt satisfait de notre travail.

II. Etape A :

On peut d'emblée dire que cette partie nous a donné une idée claire du reste du projet, c'était vraiment trop guidée, la chose qui nous a encouragé à bien comprendre la bonne utilisation d'ANTLR via l'exemple de la calculette.

Dans un premier temps, on a eu du mal à comprendre à quoi sert exactement le "setLocation", mais à un moment donné on bien compris ce qu'il faut faire, même le parser de la partie Objet, on l'a fait durant une journée.

III. Etape B : BENSRHIER Nabil

Lors de cette partie, j'ai eu l'opportunité de voir un exemple concret par lequel j'ai pratiqué le cours : "la théorie de langage". Autrement dit, cette étape a été pour moi une expérience formatrice sur le fait d'implémenter une théorie sur le plan technique.

Il va sans dire que je suis le seul qui a travaillé sur cette partie, ce qui nécessite une concentration sérieuse pour prendre de bonnes décisions.

Il m'a fallu au début de lire tout le poly pour avoir une vision globale sur le projet, comme cela j'ai pu faire une liaison entre la sortie de la "partie A" et l'entrée de la "partie C".

Dans cet ordre d'idées, cette partie ne pose aucun problème technique, c'était juste une question de compréhension des environnements et comment on doit les implémenter sous forme d'une liste chainée.

De plus, j'ai eu du mal à écrire manuellement des tests, c'est pour cela j'ai écrit un programme python qui génère tests aléatoires, c'était juste pour la partie sans Objet, la chose qui m'a vraiment aidé à détecter pas mal de bugs.

Finalement, j'ai réussi à coder un compilateur "contextuellement fiable", qui répond au cahier de charges et qui génère une bonne entrée pour la partie C.

IV. Etape C : ABOUELOULA Ayman et BENCHEKROUN Omar:

Nous avons commencé la partie C sans objet dans la deuxième semaine en parallèle avec la partie B , nous avons essayé d'avancer le plus vite possible pour laisser plus de temps pour la partie objet qui nécessite une charge horaire importante, donc on a partagé les tâches entre les deux membres , par exemple Ayman s'est chargé des opérations arithmétiques , la déclaration des variables et l'initialisation tandis que Omar s'est chargé des opérations booléennes , while et le ifthenelse.

Ainsi juste après le rendu intermédiaire on a commencé la partie objet ,cette partie était assez compliquée donc on a décidé de procéder par incréments en implémentant au début un compilateur pour des classes sans Fields ensuite on a implémenté un compilateur pour des méthodes sans paramètres et enfin on s'est focalisé sur le restant du compilateur

Cette partie a nécessité plus d'effort et de temps mais on a pu respecter les dates et on a pu rendre un compilateur sans bug.

V. Extension: JEDOUI kacem et YAGOUTI Redouane

Le choix de l'extension TRIGO était une opportunité d'explorer les algorithmes permettants l'implémentation des fonctions les plus utilisés de la classe Maths de JAVA, le travail de documentation était intéressant et on a beaucoup appris au sujet des flottants et leurs quantifications.

On a commencé le travail bibliographique en début de la deuxième semaine où on a essayé d'implémenter l'algorithme de Cordic, Taylor, mais en avançant le travail et en comparant les résultats, on s'est vite rendu compte que mis à part certaines valeurs particulières ces algos avaient des limitations, d'où le choix d'intégrer l'algorithme de Chebyshev en plus, ce qui résume un peu la manière dont on a choisi d'aborder l'extension: Une

complémentarité entre la recherche et l'implémentation/vérifications puis on valide ou pas l'utilisation de cet algorithme.

Pour des raisons d'optimisations, on a eu recours à des algorithmes de réductions d'erreurs, surtout pour essayer de calculer ces fonctions sur des flottants supérieurs à 2**π.**

L'implémentation en deca a été la partie la plus délicate de cette partie, on a essayé de relever les erreurs rencontrés surtout dans la partie C à Omar et Ayman, on a pu corriger certains défauts du compilateur, mais au final, on n'a pas pu vraiment tester les fonctions implémentés en deca, les résultats en java étaient cohérents avec la théorie, les ulp augmentent dans les zones critiques des fonctions et dépassent 1, on a essayé d'améliorer les algorithmes et on a trouvé de meilleurs résultats surtout pour la fonction arctangente où on a pu élargir l'intervalle à [-1000,1000] et seuls les ulp très proche de 0 étaient plus grands que 1.

VI. Validation

Tout au long du projet, on a rédigé des tests pour les étapes A, B et C, une attention particulière a été donné sur les noms donnés au tests et leurs classifications respectives. Des scripts rédigés en python nous ont permis d'automatiser le processus de test, et donc de conserver l'intégrité du compilateur à chaque commit d'une fonctionnalité.
Les tests ont été rédigé au fur et à mesure du développement : Kacem et Redouane se sont chargés des tests de l'analyse lexicale et syntaxique, Nabil des tests de l'analyse contextuelle, et finalement Omar et Ayman des tests pour la génération du code.
La rédaction des tests était prioritairement dans l'esprit de vérifier que les fonctionnalités implémentées marchent et de trouver les éventuels bugs dans le compilateur. On a également rédigé des tests additionnels pour augmenter la couverture du code (74% dans Cobertura).
Globalement, il n'y avait pas de difficulté particulière dans cette partie, la tâche était suffisamment distribuée sur les membres de l'équipe et traitée uniformément au long du projet.

VII. RETOUR EXPÉRIENCE

Le projet Génie Logiciel était formateur pour nombreuses raisons, on cite le travail dans une équipe de 5 qui était certainement un challenge, et la gestion d'un grand projet avec une deadline assez serrée, ainsi que les éléments techniques dans le projet (théorie des langages, conception

fortement orientée objet…). La gestion du projet a évolué en fonctions des problèmes rencontrés.
La première semaine n'était pas évidente, on a essayé d'avoir une idée global sur toutes les fonctionnalités du compilateur, chose qui s'est avéré inutile par la suite.
Au fur et à mesure qu'on a pu acquérir une connaissance solide du sujet, on a pu avancer dans le développement. Ceci dit, on a trouvé globalement que la charge de travail donnée était trop énorme pour la durée imposée, ce qui nous a obligé à travailler pendant des longues heures et par conséquent diminuer considérablement la productivité et le moral de l'équipe.

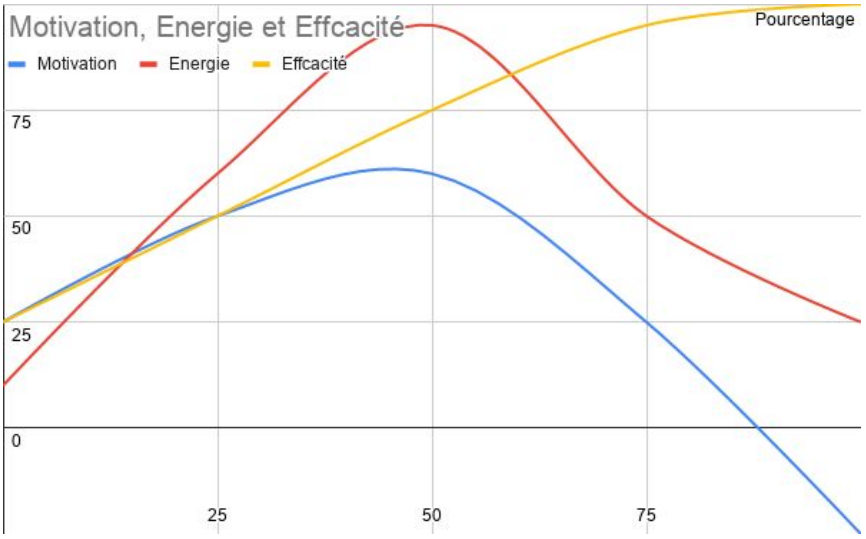

Figure : Graphe illustrant l'évolution de la motivation, énergie et efficacité du groupe gl53