

École Nationale Supérieure d'Informatique et Mathématiques Appliquées de Grenoble

Documentation de validation

Equipe 03

Benhachem Youssef
Bouhout Ilyass
Garcia Reyes Conrado Ivan
Lemrabet Soufiane
Klou Anas

Grenoble - France
28 Janvier 2021

1- Description des tests :

1.1 - Partie A :

Pour valider cette partie, nous avons commencé par créer des tests pour notre lexer en essayant d'utiliser la totalité des tokens définies dans celui-ci, vu que le lexer complet est fait dès le début.

Ensuite pour tester le parseur, on a fait des tests par partie du langage (Helloworld, Sans objet, langage complet), ce qui nous a permis de bien tester chaque partie et de faire étendre notre base de tests afin qu'elle puisse toucher à toutes les parties du langage .

Pour l'automatisation des tests de la partie A, on a fait un script *run_syntax_tests.sh* qui prend tous les fichiers sur le dossier *syntax*, et lance tous les fichiers dans le répertoire *Syntax/Valid* . Ensuite, on compare la sortie standard avec le résultat attendu.

1.2 - Partie B :

Pour la partie de l'analyse contextuelle , nous avons créé des fichiers tests dédiés pour chaque erreur, ce qui nous a permis de faire évoluer notre base de tests petit à petit et de couvrir au maximum toutes les erreurs contextuelles possibles . Pour l'automatisation des tests de la partie B on a fait un script *run_context_tests.sh* qui prend tous les fichiers sur le dossier *context*, et évaluer si l'exécution doit réussir si le test est sur le dossier validé ou échouer si se trouve sur le dossier invalid, en utilisant de la commande *test_context*.

1.3 - Partie C :

Pour cette partie , nous avons utilisé deux types de tests : des tests basiques qui testent chaque opérateur tout seul , puis des tests un peu plus avancées qui testent plusieurs à la fois en incluant une multitude de conditions et d'opérations, et des tests de performances qui visent à valider la partie C en entier en incluant tout dans un seul fichier *deca* et en testant si ça marche . Pour l'automatisation des tests de la partie C on a créé deux script *run_codegen_test.sh*, et *run_object_test.sh*, le première prend tous les fichiers *.deca* sur le dossier *codegen/valid/created* et le deuxième prend tous les fichiers *.deca* sur le dossier *codegen/valid/object*. Le script prend le

résultat attendu au début de chaque fichier et le compare avec la sortie du fichier après l'exécution de la commande decac puis ima.

2- Autres méthodes de validation :

Durant le projet, nous avons utilisé d'autres méthodes de validations que les tests , par exemple nous avons toujours une personne qui n'a pas travaillé sur le développement d'une partie qui s'occupait de faire la revue de code pour s'assurer que tout est bien fait.

- **Validation pour la partie extension :**

La validation pour la partie extension était en trois temps:

1- D'abord la validation par les test JUnit 5 pour confirmer la précision:

A l'aide de JUnit 5, on a fait une étude délicate de la précision des différentes fonctions exigées dans différentes sections de l'intervalle de définition de chaque fonction.

2- Ensuite la validation en traçant les courbes à l'aide de java.awt

Le traçage des courbes nous a permis d'avoir une vue plus visuelle et globale sur nos fonctions et aussi les comparer avec les fonctions de la bibliothèque Math de Java

3- Apres la validation à l'aide des tests sans objets en Deca

Comme notre compilateur manquait de perfection au niveau de la partie objet, on a décidé de faire des tests sans objets pour tester la compilation en Deca de l'extension. Chaque fonction principale possède donc un fichier en Deca testant ses valeurs de sortie.

3-Gestion des risques :

- **Risques liés à la modalité du travail :**

L'un des dangers les plus considérables, du fait qu'on a choisi de travailler à distance afin d'augmenter notre productivité et l'efficacité du travail, en profitant des heures hors heure de classe. Néanmoins , cette modalité peut à tout moment devenir un obstacle pour l'avancement du projet, dans le cas où les membres de l'équipe ne sont plus motivés à travailler ou ils ne prennent plus le projet au sérieux .Pour éviter un tel danger, à la fin de chaque journée les membres sont obligés de rendre la tâche qui leur est assignée , ainsi cela nous permettra de savoir si l'un des membres est moins productif que prévu avec la possibilité de basculer en mode présentiel à tout moment .

- **L'environnement du travail:**

Afin d'assurer un bon déroulement de projet en mode à distance et en cas de problème de connexion à distance aux machines de l'Ensimag (comme exemple , les machines de l'Ensimag s'éteint au bout d'un moment et cela peut engendrer une perte des tâches réalisées) nous avons décidé d'installer l'environnement de travail sur nos machines personnels .

Absence:

Afin d'assurer la réalisation des tâches, la présence pendant les réunions est indispensable.Mais, il arrive des moments où un des membre s'absente sans nous prévenir les autres, et cela peut engendrer un retard dans la réalisation des tâches,car il y avait des tâches qui dépendent d'autres et donc on n'arrive pas à bien tester nos dépôts. Pour éviter ce danger nous avons pris la décision que chaque membre doit être connecté sur "discord"(notre plate-forme de travail choisie).

- **Maladie**

Un danger important qui peut influencer le déroulement du projet et l'un des risques rencontrés dans la deuxième semaine du projet , pour limiter un tel danger nous distribuons les tâches de la personne tombée malade entre les différents membres de l'équipe selon la priorité des tâches.

- **Risques liés à une mauvaise communication**

Une mauvaise communication au sein de l'équipe conduira sûrement à un mauvais enchaînement du projet .Pour cela , on s'est mis d'accord de faire deux réunion par jour , une première le matin de fixer les objectifs de la journée et une le soir pour y discuter de l'avancement du projet et résoudre les différents difficultés rencontrés par les membres de l'équipe.

- **Un mauvais planning/Répartition des tâches**

Après l'élaboration du planning que nous avons prévu au tout début du projet, nous avons pris conscience que ce planning n'est pas réalisable et donc nous avons effectué des changements au niveau de la répartition des rôles afin de garantir le déroulement prévu des tâches. Pour répartir les rôles nous avons fait retour à la charte de l'équipe pour connaître les points forts de chaque membre et donc lui attribuer la tâche appropriée à son profil.

- **Cas de conflit entre membres de l'équipe**

Au sein d'une équipe , il est presque inévitable que des conflits et des tensions émergent , pouvoir résoudre le problème et choisir la solution qui conviennent à tout le monde n'était pas un tâche facile à gérer . En effet, pour remédier à ce problème , et en cas de conflit de deux ou plusieurs membres d'équipe sur un choix d'implémentation ou de conception d'une partie , en faisait souvent appel à une réunion d'équipe pour en discuter afin de pouvoir décider qu'elle la solution la plus optimale entre les solutions proposées .

- **Une mauvaise gestion de GIT :**

Git est un outil primordial dans le travail de groupe dans un projet aussi grand que celui-ci , mais une mauvaise utilisation de ce dernier peut causer d'énormes problèmes comme la perte de fichier ou l'écrasement de travail fait par d'autres collègues , ce qui fait qu'à partir d'un certain moment on se retrouve avec un répertoire totalement erroné .

- **L'oubli d'une date du rendu:**

Dans ce projet, plusieurs rendus existent tout au long du projet, ce qui fait que c'est très important de se rappeler des différentes dates et se préparer à l'avance , un oubli d'une date va pénaliser toute l'équipe qui se retrouvera sans rendu .

- **La mal compréhension des consignes**

il est très important de comprendre le sujet. Car la mauvaise interprétation peut affecter négativement l'implémentation et donc la qualité de rendu. Alors, on discute entre nous ce qu'on devra faire pour une étape et nous nous servons aussi du polycopié et les vidéos pour bien comprendre ce qui est attendu.

- **La perte d'information ou d'outils de travail suite à un incident**

Un risque très important que nous avons rencontré durant notre projet , est la perte d'un des membres de l'équipe son environnement de travail suite à un incident. La proposition de renvoyer les tâches faites durant la réunion journalière nous a permis de récupérer le travail fait par les membres de l'équipe et de les stockées dans un répertoire avec accès sécurisés .

- **Gestion des rendus:**

Au niveau de la gestion des rendus , les deux première parties A et B était souvent finies avant les dates des rendus que ce soit pour le print-Hello , le Sans objet ou pour l'Objet. Pourtant , on avait souvent du retard dans la partie C, ce qui a empêché la délivrance d'un rendu bien valide et bien testé . Pour la documentation et du fait que tous les membres de l'équipe partage le rôle de "Documentation Writer", ils participent tous à la documentation des différentes parties au fur et à mesure avec le développement et la génération des différents types de tests.