



PROJET GÉNIE LOGICIEL

2023/2024

Bilan de gestion d'équipe et de projet

Antoine Borget
Pierre Lin
Anass Haydar
Yassine Safraoui
Jihane Zemmoura

Groupe GL-47

Table des matières

1	Introduction	2
2	Charte de l'équipe	2
2.1	Matrice SWOT	2
2.2	Rôles et responsabilités dans l'équipe	2
2.3	Communication et outils de programmation	3
3	Planning	3
3.1	Diagramme de Gantt	3
3.2	Description du planning prévisionnel	4
4	Historique du projet	4
4.1	Développement du compilateur	4
4.2	Développement de l'extension	4

1 Introduction

Ce document est un bilan de notre gestion du projet et du travail en équipe dans le cadre du projet Génie Logiciel. Dans ce bilan, nous allons détailler les techniques de gestion de projet et d'équipe que nous avons utilisé afin de promouvoir un climat propice à la collaboration efficace, à la communication transparente et à l'atteinte des objectifs fixés. On détaillera le contenu de notre charte de travail en mettant en lumière des aspects cruciaux tels que l'auto-évaluation individuelle à travers la matrice SWOT, la répartition des rôles et responsabilités, la communication, les outils de programmation utilisés. On présentera ensuite notre planning prévisionnel et son évolution au fil du temps grâce à un historique du projet où on présentera notre organisation effective.

2 Charte de l'équipe

Afin d'organiser notre travail en tant qu'équipe de 5 nous avons défini au début du projet une **charte de travail en équipe**. Cette charte définit les règles de fonctionnement que nous avons mis en place collectivement pour rendre le plus efficace possible notre travail et pour développer une meilleure cohésion entre les membres de l'équipe. Nous allons développer dans les parties suivantes le contenu de cette charte.

2.1 Matrice SWOT

Dès la première séance du projet, nous avons rempli individuellement une fiche d'auto-évaluation où on a évalué nos compétences personnelles et techniques. Nous avons ensuite discuté de nos réponses avec les autres membres de l'équipe afin de pouvoir établir la matrice SWOT de l'équipe.

2.2 Rôles et responsabilités dans l'équipe

Après avoir construit notre matrice SWOT et avoir discuté les compétences de notre équipe, ses points forts et ses points faibles, nous avons pu définir un rôle pour chaque membre de l'équipe :

- Chef d'Équipe : Anass
- Responsable Optimisation : Yassine
- Responsable Développeur : Pierre
- Responsable Théorie de Langage : Antoine
- Responsable Qualité du Code : Jihane

Ces rôles ne définissent pas les tâches exclusives de chaque membre mais sont plutôt des statuts qui permettent aux autres membres de l'équipe de s'adresser à la personne chargé d'un aspect du projet en cas de problème.

2.3 Communication et outils de programmation

Afin d'assurer une communication optimale entre les membres de notre équipe, nous avons créer un groupe WhatsApp où on discute continuellement de l'avancement du projet, des bugs rencontrés ainsi que des éventuelles séances de codage en groupe qu'on effectue dans les salles de l'Ensimag.

Le projet est développé sous le logiciel de gestion de versions décentralisé Git, et la programmation de notre compilateur s'est faite avec l'IDE IntelliJ qui permet la programmation par pairs grâce à l'extension collaborative CodeWithMe.

Pendant la première semaine, nous avons eu quelques problèmes de conflit de merge avec Git que nous avons dû résoudre à la main. Cependant, après la première semaine, nous avons eu une meilleure vue d'ensemble du projet et chacun savait ce qu'il devait faire. À partir de là, nous n'avons eu aucun problème de merge Git nécessitant notre intervention, Git pouvait faire le merge tout seul automatiquement. De plus, nous n'avons utilisé qu'une seule branche Git tout le long du projet, et, comme les tâches étaient bien découpées, cela n'a pas posé de problème.

3 Planning

3.1 Diagramme de Gantt

Nous avons opté pour l'outil Gnome Planner pour établir le planning prévisionnel de notre projet sous forme d'un diagramme de Gantt.

Le [diagramme de Gantt](#), couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps. Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin.

Ce diagramme permet donc de visualiser d'un seul coup d'œil :

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche
- Le chevauchement éventuel des tâches, et la durée de ce chevauchement
- La date de début et la date de fin du projet dans son ensemble

3.2 Description du planning prévisionnel

Le planning du projet se trouve dans le répertoire /planning sous le nom Planning.pdf. On avait prévu que le compilateur serait terminé le 8 Janvier pour que les deux dernières semaines du projet soient consacrées à l'extension, à des tests plus approfondis et à la rédaction des rapports. Le planning comporte également les dates des suivis et des rendus en forme de Milestones afin qu'on situe facilement notre avancement par rapport aux dates importantes du projet et les flèches traduisent les dépendances entre les différentes tâches.

4 Historique du projet

Le planning effectif du projet se trouve dans le répertoire /planning sous le nom Realisation.pdf.

4.1 Développement du compilateur

Nous avons suivi une démarche incrémentale dans la définition des tâches où chaque incrément correspond à une « fonctionnalité » du point de vue de l'utilisateur. Dans le cas de notre compilateur, un tel incrément correspond à un « sous-ensemble cohérent » du langage Deca à savoir les langages Hello World, sans objet, essentiel et complet. Pour chaque langage, un compilateur complet est développé et testé suivant les étapes A, B et puis C. La conception du compilateur d'un langage n'a été entamée que quand le sous-ensemble d'avant a été terminé.

Contrairement au planning prévisionnel, le compilateur n'a été achevé qu'au 14 Janvier. La conception et le codage de l'étape C pour le langage essentiel ont été bien terminés le 8 janvier mais la validation de cette étape avec des tests appropriés a nécessité plus de temps que prévu. Entre temps, des bugs dans l'étape A et B ont été identifiés et corrigés juste avant le rendu intermédiaire.

4.2 Développement de l'extension

Le développement de l'extension a commencé comme prévu le 8 Janvier en dépit du retard du compilateur. Les premiers jours consistaient à faire des recherches approfondies sur le mode d'affichage de la GameBoy et ses mécanismes de fonctionnement. Ensuite, une fois que le choix du compilateur intermédiaire RGBDS a été fait, nous avons entamé la construction de la bibliothèque GameBoy ainsi que la totalité de la génération de code pour le nouveau langage assembleur. En parallèle, le jeu *Snake* a été développé et testé en langage Deca.