

# Rapport Lot B

Gautier Poursin, Théo Lazzaroni, Bastian Padiglione

26 avril 2020

## Introduction

Dans cette partie, nous avons pour but de rendre le fichier **main.c** fonctionnel. Pour cela, il faut implanter toutes les fonctions créées dans le lot A.

## Taches B2 et B3

Notre objectif était de créer une API permettant de gérer les données à l'aide de listes, piles et files. À l'origine il était prévu que nous puissions avoir plusieurs listes contenant chacune des données de types différents. De ce fait, afin d'éviter de dupliquer du code, nous avons décidé que l'API marcherait avec des types génériques. Les 3 types de structures sont implémentées de manière doublement chaînée, et utilisent toutes les trois le type **Node** comme maillon.

Ensuite, face à la difficulté à utiliser des types génériques dans le reste du code, nous avons décidé de rajouter une surcouche concrète aux types génériques, en encapsulant les structures génériques dans des structures concrètes. Cela permet de facilement intégrer de nouveaux types concrets sans avoir à toucher au moteur générique. Nous avons donc gardé comme fichier pour les types concrets : **structure.c** et **structure.h**. Les types génériques sont quant à eux dans les fichiers **structureEngine.c** et **structureEngine.h**.

## Tache B4

Nous avons dû modifier **carte.h** ici. En effet, nous lui avons rajouté toutes les fonctions permettant de réaliser différentes actions en fonction de la carte jouée.

Nous avons ensuite rajouté des fonctions permettant la création des cartes ainsi que des fonctions permettant de récupérer le coût, l'ID d'une carte...

Enfin, nous avons ajouté deux fonctions pour libérer l'espace mémoire alloué à une carte ainsi qu'une fonction permettant d'afficher une carte.

Ensuite, le codage de **carte.c** n'a pas été très compliqué. Il est composé principalement de 2 fonctions:

-**personalCardEffect** qui recense toutes les différentes actions que peuvent faire les cartes personnels. Cette fonction fait appel à plusieurs petites fonctions qui réalisent les différentes actions. On sait quelle action réaliser via un **switch** sur l'ID de la carte jouée.

-**actionCardEffect**: cette fonction est identique à la précédente, mais réalise les actions **des cartes actions**.

Enfin, la 3eme grosse fonction est la fonction **find**, qui elle permet de récupérer toutes les informations d'une carte grâce à son ID. La encore on fait appel à la méthode **switch**.

## Tache B5 et B6

Pour cette partie, nous avons décidé de rajouter une fonction qui permettent d'initialiser le deck, c'est à dire que cette fonction **Stack \*createDeck()**; renvoie un deck, représenté par une pile. Pour créer ce deck, nous avons d'abord créer une fonction **List \*allCard()**, qui permet d'avoir une liste de toutes les cartes disponibles pour jouer. La création du deck repose sur le fait qu'on choisit aléatoirement une carte de la liste pour on l'ajoute au dessus du deck.

Nous avons aussi créer une fonction permettant d'initialiser la main, c'est à dire qu'avant une partie, chaque joueur aura 2 cartes en main, les 2 cartes qui se situent en haut du deck initialement créer, par la fonction **List \*initializeMain(Player playerA)**; La main est une liste puisque on peut jouer n'importe quelle carte de la main.

Pour l'initialisation, nous avons aussi créer une fonction **Stack \*initializeDiscard()** et **Queue \*initializePersonnel()** pour initialiser la défausse et l'espace personnel. Toutes ces fonctions assemblées permettent notamment l'initialisation du plateau avec la fonction **Board newBoard()**. Cette fonction a d'ailleurs posé énormément de problème puisqu'elle a engendré beaucoup d'erreurs de segmentation. Cela venait d'une mauvaise utilisation du **malloc**.

Ensuite, nous avons créer des fonctions permettant par exemple d'avoir le nombre de tour(**int getLap(Board board)**), ou même le nombre de PE disponibles (**int availablePE(Player player)**).

Les fonctions d'initialisations ont été la partie la plus dur de cette partie. Elles ont causé énormemement de problème de segmentation. Une fois ces problèmes réglés, le plus dur à été de ne pas se mélanger entre toutes les fonctions et les différents élément de joueurs.

## Tache B7

Cette tâche est essentielle pour le dialogue entre utilisateur et ordinateur. En effet, l'utilisateur doit donner des informations à l'ordinateur pour qu'elle exécute ses ordres. Sans cette interface, il n'y a aucune communication entre utilisateur/ordinateur.

La partie la plus dur dans cette tâche a été la fonction **printBoard**. En effet, il a fallu créer en plus des fonctions pour afficher des listes de cartes, des files... Ensuite, il a fallu bien gérer les **scanf** pour bien renvoyer le nombre rentré par l'utilisateur et non son adresse par exemple.

C'est aussi dans ce fichier qu'il y a toutes les options d'affichages et toutes les informations que l'on donne aux joueurs à chaque fin de tour.

## Tache B8

Phases de test

## Conclusion

Durant ce projet, nous avons remarqués l'importance d'une cohésion d'équipe et de la communication entre chacun. En effet, il était nécessaire de savoir exactement ce que chacun faisait et comment chacun avançait. Pour cela, nous avons du apprendre à utiliser GitLab, qui était inconnu pour Théo et Gautier.

Concernant le projet en lui même, il fallait faire très attention entre les points de Développement, de durabilité... Nous avons longtemps confondu développement, durabilité et énergie et donc cela à énormément compliqué la mise en code. Les explications étaient ambiguës, une démonstration du jeu aurait pu permettre de mieux comprendre le jeu. Par exemple, notre structure "Card" ou "Player" a changé entre le lot A et le lot B, de même pour le main.c. Il aurait fallu mieux appréhender le sujet dès le départ et savoir exactement de quoi on parlait dans le jeu et cela aurait pu simplifier le code.

Ensuite, nous avons du aussi apprendre à travailler à distance. Cela a été assez difficile au début puisque c'était une chose nouvelle et il a fallu savoir expliquer ses erreurs à distance. Concernant l'enseignement à distance, il s'est bien passé malgré ce confinement. Le référent était bien et nous a laissé plus de temps pour le lot B car nous nous sommes retrouvé à 3, ce qui nous était nécessaire pour nous pour finir le lot B.