

Rapport lot B

Groupe 17

Léanna JI, Théo SZATKOWSKI, Luca MICCICHE, Loan GODARD

Impression générale :

Nous nous sommes réparti assez vite les tâches et nous avons gardé une bonne communication au sein du groupe. Malgré le confinement, nous n'avions pas rencontré de problème de communication et nous sommes tous satisfaits du travail rendu.

Dans l'ensemble le lot B s'est bien passé, les tâches ont été faites dans les temps et bien réalisées

Problèmes rencontrés

Nous avons rencontré plusieurs problèmes lors de ce lot B :

Les structures

Pour les piles et les files, nous avons commencé par fonctionner avec des pointeurs et nous devions alors entrer des adresses en argument pour modifier les piles et files. Nous avons souhaité modifier cela de sorte à ce que l'on puisse passer directement les files et piles en argument pour pouvoir les modifier sans passer par les adresses. On a réussi à faire cela pour les piles que l'on passe directement en argument des fonctions pour les modifier. Cependant nous n'avons pas réussi à le faire pour les files qui sont en fait des listes chaînées car nous avons eu des problèmes pour les modifier. En effet les files sont des pointeurs vers la valeur suivante mais si on veut les modifier en place sans passer par les adresses on doit passer par un pointeur de pointeur ce qui nous a posé problème. C'est pourquoi nous avons choisi de garder les files comme simple pointeur vers la valeur suivante et donc pour modifier les files on doit passer des adresses de pointeur en argument.

Les effets des cartes Personnel et Action:

On ne voulait pas gérer les effets 1 par 1 et nous avons voulu gérer les cartes de façon à ne pas rencontrer de problème en ajoutant une nouvelle carte ou de changer des effets lors du lot C par exemple. Nous avons rencontré beaucoup de difficulté à essayer de gérer à la fois les effets des cartes Personnel et les cartes Action d'un seul trait. En effet, ces effets ne sont pas tout à fait pareils. Tandis que les cartes Personnel peuvent avoir plusieurs effets où il s'agit uniquement de retirer ou d'ajouter de DD ou autre, les cartes Actions peuvent avoir des effets très impactants et il ne s'agit pas uniquement d'ajouter ou de retirer des points. De plus, les cartes Personnel restent sur le plateau de jeu tandis que les cartes Actions vont directement dans la défausse lorsqu'on les joue.

Face à ce problème, nous avons dû faire des compromis :

-Les cartes Action devront être traité 1 par 1 car elles ont des effets très différents l'une des autres.

-Les cartes Personnel cependant ont 12 effets max possibles et chaque carte attribue juste des points entre 0 à 2 sur ces 12 effets. Ainsi, nous avons eu l'idée de créer un tableau d'effet (AE1, AE2..., DR, E) et on aura qu'à récupérer la valeur des points de chaque effet d'une carte. Ensuite, on a plus qu'à créer une fonction pour chaque effet possible et à les appeler en donnant en paramètre le nombre de points que la carte a pour chaque effet. Nous n'appelons pas les fonctions si la carte a 0 point pour ces effets-là.

L'interface

En ce qui concerne le l'interface, le but était de pouvoir lire toutes les informations majeures sur le Terminal. Dans un premier temps, celui-ci était en Anglais, puis sur décision commune, nous avons décidé de le faire en Français pour plus de lisibilité.

Pour les deux fonctions utiles au jeu, nous avons essayé de prendre en compte tous les cas possibles, afin que l'utilisateur puisse continuer de jouer même s'il se trompe. (s'il entre une lettre plutôt qu'un chiffre, un mauvais Id etc.). Comme expliqué dans la notice, nous avons essayé de rendre l'utilisation du jeu simple, et rapide à prendre en main.

Les tests :

Nous avons surtout été bloqué pour la mauvaise gestion des mallocs et des pointeurs. Il s'agit en général d'une erreur d'inattention mais qui nous a fait perdre beaucoup de temps.

Une première erreur a été d'écrire « `malloc(sizeof(sizeof(struct card)))` », une erreur d'inattention et qui a été difficile a remarqué pour débbugger notre programme.

Une seconde erreur a été de changer un nombre par un DEFINE mais d'avoir mal définie ce DEFINE. Cela ne posait pas de problème pour 75% du temps et donc l'erreur a été découverte bien plus tard. Ce n'était pas dur de retrouver l'erreur mais pas évident.

Les 25% restants c'était un autre malloc qui écrivait sur notre premier malloc qui avait une taille réservée trop petite et donc nous avions des résultats incohérent pour une partie du premier malloc et des résultats cohérents pour la seconde partie du malloc.

Lors de certaines fonctions comme la fonction pour pop un élément d'une Pile, un oubli de changer la taille de la pile était également présent. Ce qu'il rendait la pile théoriquement infinie mais au bout d'un moment il y avait bel et bien plus aucun élément dans celle-ci.