
Rapport sur le projet de Programmation Impérative 2020/2021

Réalisé par
LINARD Jules

Dans le cadre du cours
De Programmation Impérative

Travail présenté à
Guillaume BUREL
Professeur de programmation impérative
À l'école nationale supérieure d'informatique pour
l'industrie et l'entreprise

ENSIIE
A rendre le 04 janvier 2021

Introduction

Dans le cadre du projet de programmation impérative, j'ai dû réaliser un jeu de plateau dont les règles sont assez simples :

- Initialement, le plateau est vide.
- Chaque joueur joue à tour de rôle.
- À chacun de ses tours, le joueur pose un jeton de sa couleur sur une des cases du plateau, en l'empilant sur les jetons déjà présents le cas échéant.
- Si les deux jetons au sommet de la pile sont de la même couleur, la case est activée :

- . Les deux jetons sont retirés de la pile ;

- . Des jetons de la couleur du joueur sont ajoutés sur les quatre cases voisines, à condition qu'elles n'aient pas déjà été activées à ce tour du joueur ;

- . Les jetons ajoutés sur les cases voisines peuvent activer celles-ci, si elles avaient aussi un jeton de cette couleur en haut. Des jetons sont alors ajoutés sur les cases de leurs voisines qui n'ont pas déjà été activées ce tour, et ainsi de suite...

- La partie se termine lorsque toutes les cases sont remplies et le joueur gagnant est celui qui possède le plus de jetons au sommet des piles.

Choix relatifs au projet

Problèmes et solutions

Ce projet laissait le choix sur les méthodes utilisées et la façon d'implémenter le jeu. Ainsi, je vais exposer ici les choix que j'ai effectué afin de réaliser au mieux ce projet.

Il m'a fallu tout d'abord implémenter les piles. En effet, j'ai voulu considérer mes cases de plateau comme étant des piles, puisque les jetons pouvaient s'empiler sur chacune d'entre elles. Pour se faire, j'ai pris la décision d'implémenter mes piles comme si elles étaient des listes chaînées dans lesquelles on ne pourrait accéder qu'à l'élément situé "au sommet" de cette liste. En plus des fonctions classiques telles que tester si la pile est vide, lui retirer ou ajouter un de ses éléments par exemple, j'ai dû également implémenter une fonction permettant de comparer le sommet de la pile avec un certain élément, ce qui me sera très utile lorsqu'il faudra tester l'activation ou non d'une case.

Dans un deuxième temps, j'ai dû implémenter une nouvelle structure pour mon plateau. M'inspirant des matrices, qui sont des tableaux 2D d'entiers ou de flottants, j'ai opté pour un plateau 2D dont les cases seraient des piles. J'ai donc tout d'abord créé une fonction me permettant d'initialiser ce plateau en allouant la place nécessaire en mémoire. C'est à ce moment que les premiers problèmes sont apparus. En effet, il fallait tout d'abord prendre en

compte l'effet "cascade" des activations de case qui pouvaient s'activer suite à une autre activation. Après réflexion, j'ai décidé d'implémenter une fonction récursive. En effet, cela me semblait être le meilleur choix afin de prendre en compte cet effet "cascade". Par la suite, un autre problème est apparu : une case ne pouvant s'activer qu'une seule fois par tour, il fallait retenir les activations de cases durant un tour. Pour se faire, j'ai décidé de créer un tableau 2D d'entiers rempli de 0. Ces derniers se transformeront en 1 lors de l'activation d'une case ce qui l'empêchera de s'activer une nouvelle fois.

Dans un dernier temps, il a fallu implémenter la fonction principale du projet, qui rassemblait toutes les autres et les utilisait. J'ai respecté du mieux que je pouvais la mise en page demandé par le projet, que ça soit au niveau de l'interaction avec l'utilisateur ou encore pour l'affichage du plateau de jeu et des piles. Lors de la compilation, quelques erreurs sont apparues notamment à cause des interfaces que j'importais : elles n'étaient pas dans le bon ordre par rapport à leurs dépendances. De plus, j'ai dû modifier l'emplacement des définitions de mes structures : elles étaient initialement dans mes fichiers.c et sont désormais dans les fichiers.h .

Suite à ces modifications, le jeu fonctionnait normalement et les diverses parties que j'ai effectuées en guise de test ont été concluantes.

Conclusion et limites du projet

Pour conclure, comme dit précédemment, ce projet est fonctionnel et permet de jouer de façon convenable. Cependant, des pistes d'améliorations sont possibles et des limites existent :

- Lors de la saisie de la case, il faut impérativement un espace entre l'abscisse et l'ordonnée de la case, sinon le jeu ne place pas correctement le jeton.
- Lors de la confirmation pour le placement du jeton, le programme n'accepte que les réponses "oui" ou "o" et non toutes les réponses commençant par "o".
- La fonction récursive activation plateau est un peu lourde et pourrait être simplifiée.