

# Programmation C avancée

## TP 4 - Jeu de Taquin

### Introduction

Ce TP de programmation consiste à créer un jeu du taquin.

Dès le lancement du jeu, une image se découpe en plusieurs cases, et celles-ci sont toutes mélangées. Une seule case n'est cependant pas affichée, et est représentée par une case noire : il s'agit de la case située tout en bas à droite de l'image.

Le joueur doit ainsi replacer chaque case de l'image à son endroit initial, pour ainsi reformer l'image d'origine.

### Mode d'emploi

Dans un terminal, se rendre dans le dossier du projet.

A l'aide du makefile, il suffira ainsi juste d'entrer la commande « make », et l'exécutif du jeu se créera sous le nom de « Taquin ». Pour lancer le jeu, il faudra entrer la commande « ./Taquin ».

Une fois le jeu lancé, plusieurs menus s'afficheront un à un, dans lesquels l'utilisateur entrera les choix qu'il va faire.

Le 1<sup>er</sup> menu correspond aux dimensions du jeu. Le joueur a 3 possibilités : 2x2, 4x4, ou 8x8. Celui-ci devra entrer respectivement 1, 2 ou 3 selon son choix.

Le 2<sup>nd</sup> menu demandera au joueur combien de fois va-t-on mélanger les cases du plateau : 20, 60 ou 120 fois. Il est important de noter que X mélanges ne signifie pas que le joueur devra faire X mouvements pour résoudre le jeu : lors du mélange des cases, il se peut qu'on réalise des mouvements qui s'annulent. Le programme va cependant toujours donner un plateau à résoudre : effectivement, si le programme propose un plateau déjà résolu, il remélange celui-ci.

Le 3<sup>ème</sup> menu demande au joueur de choisir une image parmi les 4 qui lui sont proposées.

Enfin, le dernier menu correspond à un bref rappel des règles, ainsi que les commandes. Effectivement, le jeu peut se jouer avec les touches directionnelles : le joueur devra ainsi faire → s'il veut que la case à gauche de la case noire remplace celle-ci, etc.

Cependant, on peut également jouer avec le clique gauche de la souris. Le joueur devra ainsi cliquer sur une case adjacente à la case noire du plateau pour échanger ces 2 cases.

Pour lancer le jeu, le joueur devra ensuite cliquer sur le bouton « Lancer ».

Durant la partie, on affiche le nombre de coups joués par le joueur, mais également le temps de sa partie. Il dispose d'un temps et d'un nombre de coup illimité pour résoudre le jeu.

Le joueur a également la possibilité de recommencer la partie dès que bon lui semble en cliquant sur le bouton « Recommencer ». Dans ce cas là, on remélange le plateau de la même façon qui a été faite au début de la partie, et le temps et son nombre de coups joués sont remis à zéro.

Enfin, le joueur a également un bouton « Quitter » qui, de la même façon que la croix de la fenêtre, interrompt la partie en cours et quitte la fenêtre.

Lorsqu'il termine sa partie, le joueur a un message de victoire et une fenêtre qui lui indique dans combien de seconde la fenêtre va se fermer de manière automatique (20 secondes).

## Architecture

Le programme a été découpé en 4 modules :

- *Le main, Main.c*

Contient le code du jeu, et la façon dont le jeu est géré en temps réel.

- *Affiche\_Image.c/Affiche\_Image.h*

Ce module va se charger d'afficher l'intégralité de l'interface du jeu : le plateau, le quadrillage, l'image d'origine à droite, l'interface du jeu (le menu en bas du plateau, i.e. le nombre de coups joués, le temps et les boutons « Recommencer » et « Quitter »). On a également les fonctions qui contrôlent si le joueur clique sur les boutons « Recommencer » ou « Quitter ».

De cette façon, on ne mélange pas le code qui se charge du fonctionnement du jeu, et de la partie graphique du jeu.

- *Menu.c/Menu.h*

Un peu identique au module différent, *Menu* se charge d'afficher les différents menu dans lesquels le joueur choisira les options de son jeu.

Cependant, à la différence d'*Affiche\_Image*, on récupère les données que le joueur va nous donner pour nous permettre de créer le plateau selon les options que celui-ci nous aura donné.

On récupérera les données concernant la taille du plateau, le nombre de mélange, et le choix de l'image.

Le module se chargera enfin d'afficher un bref résumé du jeu et de ses commandes, et créer un bouton « Lancer » sur lequel le joueur doit cliquer pour démarrer le jeu.

- *Plateau.c/Plateau.h*

Ce module se charge de l'intégralité des commandes du plateau : sa création, ses commandes, le mélange des cases, mais également les conditions de victoire, ou bien la réinitialisation du jeu (la fonctionnalité « Recommencer »).

De la même façon que le module *Affiche\_Image*, ce module se charge de l'intégralité du fonctionnement du jeu : ainsi, si on a un bug lors de la partie, on vérifiera toujours ce module.

Naturellement, le *Main* va dépendre des modules *Affiche\_Image* pour afficher l'image, *Menu* pour afficher le menu, et *Plateau* pour les fonctionnalités du jeu.

*Affiche\_Image* va également dépendre de *Plateau* : en effet, on récupère très régulièrement les données du plateau, par exemple la position actuelle des cases mélangées, ou bien les dimensions du plateau.

La représentation des fonctions et leur fonctionnement se trouve tous dans les headers, les expliquer ici serait donc redondant et sans intérêt.

## Choix d'implémentation

Comme expliquer dans la partie précédente, il m'a paru nécessaire de séparer le programme de sa partie graphique et sa partie technique : mélanger les deux n'aurait pas faciliter le débogage, ni la représentation de chaque fonctionnalité du projet.

J'ai également été décidé séparer les menus qui s'affichent avant le projet, et la partie graphique en temps réel du jeu.

Effectivement, même s'ils représentent effectivement tous les deux des parties graphique, l'un se charge du jeu en temps réel, alors que l'autre représente les données que le joueur va faire avant le jeu, avant même que celui-ci n'ait encore commencé.

Une fonctionnalité « Recommencer » est également ajoutée : celle-ci permet au joueur de reset son plateau à n'importe quel moment de la partie. Le plateau sera remélangé strictement de la même façon qu'avait été fait son plateau à l'origine.

Il m'a paru important de rajouter cette fonctionnalité, puisque durant une partie, il se peut qu'un joueur ne soit pas satisfait des choix qu'il a fait, et souhaite donc recommencer sa partie.

Le joueur a également la possibilité de jouer avec les touches directionnelles, mais également avec la souris. Effectivement, il peut être plus instinctif de jouer avec la souris plutôt qu'avec les touches directionnelles, d'où l'ajout de cette fonctionnalité.

Enfin, il est important de parler de la structure du plateau, qui n'est pas la même que celle proposée par l'énoncé.

J'y ai ajouté plusieurs champs :

- **nb\_col** et **nb\_lig** : pour offrir plusieurs options de dimensions au joueur, on ne peut pas laisser les variables **nb\_col** et **nb\_lig** en tant que variables globales : il aurait été moins évident de les modifier par la suite dans le projet.  
Il me paraissait donc naturel de les implémenter dans la structure plateau, puisque de cette façon, au même moment qu'on initialise le plateau, on initialise également ses dimensions.
- **Carre bloc[NB\_COL][NB\_LIG] → Carre \*\*bloc** : Les dimensions du plateau n'étant plus fixe, on a besoin de les changer selon la taille entrée par l'utilisateur. On va ainsi allouer la place nécessaire pour le tableau bloc selon les dimensions entrées par l'utilisateur.
- **coups\_joues** et **time** : Ces fonctionnalités n'étant pas implémentées de base dans l'énoncé, les rajouter dans la structure facilite leur modification en temps réel.

On peut également discuter des choix graphiques qui ont été faits, comme l'ajout de l'image d'origine qui est affichée pour permettre au joueur de s'y référer durant la partie.

## Expérience d'apprentissage

Ce projet m'a essentiellement permis de re-maîtriser la librairie MLV, librairie que j'avais déjà utilisée lors de projets durant ma deuxième année de licence. Cependant, je n'ai pas utilisé de nouvelles mécaniques ou fonctionnalités de la librairie par rapport aux projets que j'ai pu faire l'an passé.

De la même façon, j'ai rencontré très peu de difficultés techniques grâce à l'expérience que j'ai acquise pendant ma deuxième année de licence.