

Di Giorgio Mattéo
Saleh Suhayb
Chassery Alexis

Compte-rendu Projet IF2B

Introduction :

Lors de ce projet, notre but est de créer un jeu appelé 2048 avec plusieurs mode de jeu tel que « Duo » ou « Puzzle » et une grille variable allant de 4x4 au 9x9. Nous verrons comment nous l'avons crée, les problèmes que l'on a rencontré, et les concessions que nous avons du faire.

Notre programme est décomposé en deux parties, la partie graphique et la partie logique. La partie logique étant toute les fonctions telles que la comparaison de deux valeurs, ou la sauvegarde. Et la partie graphique concerne les fonctions d'affichage dans la console.

I) Initialisation du jeu

Pour débiter le jeu nous allons d'abord demander a l'utilisateur de choisir parmi les 3 mode de jeux « Classique », « Duo » et «Puzzle » en écrivant 1,2 ou 3. On va contrôler que l'utilisateur écrit bien ce que l'on veut avec un message d'erreur. Ensuite on vient demander au joueur la taille du tableaux avec laquelle il veut jouer allant de 4x4 à 9x9, ceci s'applique uniquement pour les mode « Classique » et «Duo ». On va aussi contrôler la taille du

```
while (mode<1||mode>3) { // Si le joueur accepte de jouer on lui demande le mode de jeu qu'il veut
    printf("Veuillez choisir la mode de Jeu \n1 : mode Classique\n2 : mode Duo\n3 : mode Puzzle \n");
    scanf("%d",&mode);
    if (mode<1||mode>3){
        printf("Saisie incorrect, Veuillez resaisir ! \n");
    }
}
if(mode!=3){// Si le mode de jeu n'est pas le puzzle alors on lui demande la taille du tableaux.
    while (n<4||n>9) {
        printf("Veuillez choisir la taille du tableau (entre 4 et 9) :");
        scanf("%d", &n);
        if (n < 4 || n > 9) {
            printf("Saisie incorrect, Veuillez resaisir ! \n");
        }
    }
}
fflush(stdin);
jeu(n,mode);
```

tableaux avec un message d'erreur si nécessaire. Nous allons initialiser un tableaux de la taille renseigné.
Le jeu va pouvoir commencer !!

II) Déroulement du jeux

Pour commencer il va falloir afficher le tableaux et le score, pour un tableaux de taille, N compris dans les naturels, NxN, on va afficher le score, faire un saut de ligne puis avec deux boucle on va afficher chaque élément du tableaux en sautant une ligne tout les n élément.

Voici un rendu avec le morceau de code correspondant :

```
void affiche(int n, int T[n][n], int*Score){
    /* La fonction affiche, print dans la console le terrain de jeu du 2048*/
    printf( format: "Score Actuel : %d\n", *Score);
    for (int i =0; i<n; i++){// Nous parcourons le tableaux
        for (int y =0; y<n; y++ ) {
            if (T[i][y] == -1){// Nous avons décidé de la lettre X sera représenté par un -1 dans le tableaux.
                printf( format: "%-7c", 'X');
            } else {
                printf( format: "%-7d", T[i][y]);
            }
        }
        printf( format: "\n");
    }
}
```

```
5Score Actuel : 2
0      0      0      0      0
0      0      0      0      0
0      2      0      0      0
0      0      0      0      0
0      0      0      0      0
```

On va maintenant passer au mouvement, il est possible de faire 4 mouvement au 2048 , haut, bas, gauche et droite. Un mouvement est constitué de 2 choses, le déplacement de chiffre et la fusion des chiffres. Nous avons choisi dans notre programme de commencer par la fusion puis le déplacement.

Pour la fonction fusion on va parcourir en largeur si le mouvement est vers la gauche ou la droite ou en hauteur si le déplacement est vers le haut ou le bas.

Prenez par exemple un déplacement vers la droite, mais le principe reste le même pour tout les mouvements, on va prendre l'élément à droite de la première ligne puis le comparer aux éléments en partant vers la gauche, si on rencontre le même chiffre alors l'élément le plus à droite double son chiffre et celui rencontré est redéfini à 0. Si on tombe sur un chiffre différents on arrête la boucle. On va recommencer pour chaque élément de chaque ligne. Ainsi nous sommes certains à ce moment là qu'il n'y a plus de fusion à faire, il est donc libre de déplacer correctement les cases afin de pouvoir débiter un nouveau tour. On va ensuite appliquer la fonction déplacement. Pour le mode « Duo » la logique est la même, seulement nous appelons deux fois les fonctions Fusion et Déplacement pour les deux tableaux.

III) Mode de jeux

Il existe deux mode de jeux supplémentaires au jeu normal, le mode « Duo » et le mode « Puzzle ». Le mode « Duo » est similaire au mode classique sauf que l'on joue avec deux tableaux classique. La gestion du mouvement et de fusion sont les mêmes.

```
Score Actuel ! 4
0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0      0
0      2      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      2      0      0      0      0
Dans quel direction voulez-vous aller ? (q/d/z/s) ou 'a' pour quitter
```

Le mode «Puzzle » est un mode de jeu où le tableaux du joueur est défini à l'avance. Il est aussi possible que dans le tableau se trouve des cases « X », celles-ci ne peuvent être fusionnées ou bougées. Pour lancer une partie de Puzzle, nous ouvrons un fichier texte contenant toutes

les informations qui nous faut. Les informations sont disposé ainsi :

```
Taille 5
0|16|64|32|2
8|16|32|64|0
2|8|8|2|0
0|16|32|2|16
2|0|0|8|2
```

La première ligne avec l'intitulé « Taille », défini la taille du tableaux

qui en l'occurrence est défini à 5.

Ensuite, est définis le contenu du tableaux : Pour séparer le contenu de chaque case nous avons défini le symbole « | » comme séparateur des cases. Bien sur il faut autant de ligne et autant de colonne que la taille indiqué. Puisqu'il est impossible de mettre autre chose que des Integers, nous avons définis le chiffre « -1 » comme étant le « X ».

Le contenu de ce fichier texte :

```
Taille 5
-1|16|64|32|2
8|16|32|64|0
2|8|8|2|-1
-1|16|32|2|16
2|0|0|8|2
```

Donne :

```

Score Actuel : 0
X      16      64      32      2
8      16      32      64      0
2      8       8       2       X
X      16      32      2       16
2      0       0       8       2
dans quel direction voulez vous aller ?

```

IV) Gestion de la sauvegarde

Pour sauvegarder une partie on va stocker les informations dans un fichier texte dans un format que l'on va définir :

```

5 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2

```

On va d'abord stocker le taille du tableaux, puis on va stocker chaque élément du tableaux puis en dernier on stocke le score de la partie. Ce qui nous permet pour la lecture du fichier de créer un tableaux de la bonne taille pour la partie sauvegarde

V) Equipe

Nous avons à trois développer différentes parties du programme que nous avons mis en commun sur GitHub, où nous avons créer un dépôt commun pour pouvoir s'échanger les fichiers mis à jour.

VI) Organisations

Le fichier main.c est le fichier qui contient l'unique fonction main(), et le fichier functions.c contient toute les fonctions nécessaire au bon fonctionnement du jeu. Il est important pour plus de clarté de ne mettre aucune fonction dans le fichier main.c par soucis de modularité.

CL functions	02/01/2024 18:03	JetBrains CLion	15 Ko
CL functions	30/12/2023 21:32	JetBrains CLion	1 Ko
CL main	30/12/2023 21:32	JetBrains CLion	2 Ko

VII) Fonctions principales

Dans le fichier functions.c il existe trois fonctions principales correspondant aux trois modes de jeu possibles :

1. normal()
2. duo()
3. puzzle()

Leur structure est similaire mais il ya quelques subtilité.

La fonction normal est la fonction qui lance le jeu « basique », une grille et aucun « X », elle peut être aussi utilisé pour lancer le jeu à partir d'une sauvegarde, il suffit de donner en paramètre un tableau déjà remplis.

```
void normal(int n, int T1[n][n], int *score){
```

Parmi les paramètres :

n → la taille du tableaux

T1 → Le tableaux en lui mêmes, peut être remplis ou vierge

*score → Un pointeur qui permet d'actualiser le score avoir à le retourner

La fonction Duo est la fonction qui lance le mode de jeu « Duo », deux grilles et aucun « X ».

A savoir que le mode « Duo » ne prend pas en charge les sauvegardes, donc la fonction « Duo » ne peut pas charger de sauvegarde.

```
void duo(int n, int T1[n][n], int T2[n][n], int *score){
```

Parmis les paramètres il y en a un de plus que pour la fonction normal :

T2 → le deuxième tableaux, peut être remplis ou vierge

Et enfin la fonction puzzle qui lance le mode « Puzzle », une grille et des « X » au choix. A savoir que ce mode de jeu ne prend pas en charge les sauvegardes.

```
void puzzle(int *score){
```

Ici, le seul paramètre est le score car il est inutile de mettre un tableaux et sa taille puisque nous le récupérons dans un fichier au cours de cette même fonction.

Bien sur il était possible de récupérer les données du fichier et ensuite appeler la fonction normal() avec le tableaux pré remplis mais nous avons préféré créer une fonction appart entière.

VIII) Résultat final

Nous avons réussi à créer un 2048+ avec toutes les fonctionnalités souhaité, au regret de ne pas avoir d'interface graphique. Les fonctions de mouvement et de fusions fonctionnent, ainsi que les fonctions de sauvegarde/Lecture. En outre les trois modes de jeu fonctionnent.

Il pourrais y avoir une amélioration sur le mode Puzzle, car celui-ci ne prend en charge qu'un fichier définis dans le code source, et un Parser plus efficace et modulaire : Si le fichier contenant les informations pour le mode puzzle ne respecte pas à la lettre les règles de rédaction, le programme s'en retrouva obligatoirement altéré et ne pourra pas fonctionner correctement. Il y a aussi une améliorations à faire sur la qualité du programme qui peut être flou ou compliqué en un programme un peu plus propre et clair.

Nous avons aussi du faire impasse sur l'interface graphique.