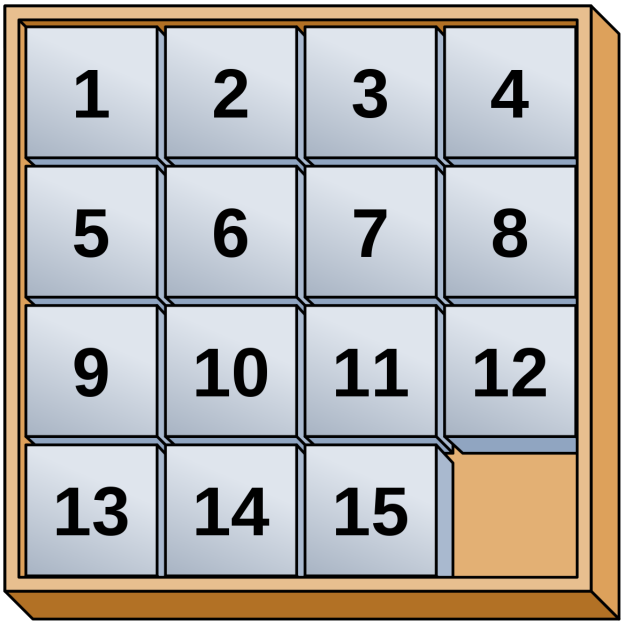
Figure 1: ISIMA

Figure 2: Université de Clermont-Ferrand

Projet de Programmation avancée en C

Réalisation d'un jeu de Taquin



CLIQUOT Théo et CHASSAGNOL Rémi
année 2020-2021

Professeurs: Vincent Limouzy
Référent TP: Nicolas Wagner

Travail à rendre pour le : 15/1/2021

Table des matières

Table des figures

1 Règle du jeux

Ce qui va suivre va permettre à n'importe quel personne de jouer au taquin. Cette section contient notamment toutes les touches et toutes les fonctionnalités du jeu. Dès le lancement du jeux on aura d'afficher le taquin, il suffira ensuite d'appuyer sur une touche pour réaliser l'action qu'on veut.

- Déplacement : Flèches directionnelle.
- Sauvegarder un taquin : touche `s`
- Charger un taquin : Pendant l'appel de l'exécutable donné en argument le nom du fichier.
- Tarquin aléatoire : Lancé l'exécutable sans autre argument (la taille sera demandé)
- Tester si le taquin est correcte : touche `t`
- ...

Les mouvements sont définis par rapport à la case vide , c'est à dire que si on à : | 5 | | 10 | et qu'on appui sur ▷, on obtiendra : | 5 | 10 | | Et inversement si on appui sur ◁, on obtiendra : | | 5 | 10 |

2 Fonctionnement

2.1 Structure du taquin

On utilise la structure suivante (renommée en texttttaquin). textttttab correspond à la postion de nos pièce par rapport à la grille (un tableau 2D),l'entier textttposCaseVide correspond à la position de notre caseVide dans le taquin (par exemple si il est en bas à droite d'un taquin de dimension 5, il sera égale à 24) et enfin l'entier texttttaille correspond à la dimension du taquin :

```
1   typedef struct taquin
2   {
3     int ** tab;
4     int posCaseVide;
5     int taille;
6
7   } taquin;
```

2.2 Sauvegarde du taquin

Dans cette fonction on sauvegarde dans un fichier de la forme `sauv txt` notre taquin actuelle. Pour cela on regarde si `sauv_0.txt` existe en l'ouvrant . Si il n'existe pas alors on peut créer un fichier représentant le taquin actuelle. Sinon on le ferme et on incrémente notre numéro de sauvegarde (`sauv_1.txt`) et ainsi de suite jusqu'à tomber sur un fichier inexistant. Notre taquin est sauvegarder de la façon suivante :

- Seul le tableau est sauvegarder puisqu'on peut retrouver les autres éléments du taquin à partir de celui-ci.
- Chaque case du tableau est séparé par un espace.
- Chaque ligne du tableau est séparé par un saut de ligne.

2.3 Déplacement

Pour le déplacement il nous suffit juste d'avoir le taquin est la direction, on regarde si en fonction de la direction que l'on veut notre case vide ne va pas déborder, si ce n'est pas le cas alors on effectue une permutation entre 2 cases (et on oublie pas de mettre à jour la position de la case vide).

2.4 création du taquin

2.4.1 Taquin aléatoire

Pour créer un taquin aléatoire selon une taille donnée, il nous suffit de créer un taquin avec un tableau comme on le veut à l'arriver et assigner à `posCaseVide` et `taille` la bonne valeur. Puis on applique un certain nombre de fois des déplacementes afin de mélanger le tableau.

2.4.2 Taquin charger

Il nous suffit de lire un fichier avec la même construction que notre fichier de sauvegarde et récupérer les valeurs de `tableau`, `taille` et `posCaseVide`

2.5 Taquin Trié

Il suffit de parcourir le tableau est de regarder si dans la case 0 on à 1, dans la case 1 on à 2 ... et pour la dernière -1.

3 L'affichage SDL

Il y a en tout cinq fonctions d'affichage pour cinq écran différents. La première fonction va afficher le message de début de jeu (et le logo), la seconde va afficher un message qui demande à l'utilisateur si il souhaite sauvegarder sa progression dans la partie. Les trois autres vont afficher les écrans de jeu de fin et le menu d'aide.

3.1 L'affichage Principal

Cette fonction est la fonction d'affichage principale, elle va afficher les différentes parties de l'image en fonction de la grille du taquin. Chaque case du tableau correspond à une certaine partie de l'image en fonction de la valeur quelle contient. Par exemple, une case qui contient 1 correspond au coin en haut à gauche de l'image. La taille de la surface de l'image qui correspond à une case de tableau dépend de la taille de la grille.

Cette fonction utilise une seule surface (sans compter l'écran) qui contient l'image et deux **SDL_Rect**, une qui sert à découper l'image en plusieurs partie et l'autre sui sert à coller chacune de ces parties sur l'écran.

3.2 Les affichages secondaires

3.2.1 L'affichage de l'écran de départ

Cette fonction permet d'afficher le petit texte qui apparaît lorsqu'on lance le programme. Elle affiche aussi le logo de DOOM en utilisant la fonction **SDL_SerColorKey** et l'option **SDL_SRCCOLORKEY** pour ne pas afficher le fond blanc du logo. Elle fait appel à la fonction attend touche pour attendre que l'utilisateur appuis sur une touche pour démarrer le jeu et donc changer l'affichage. Cette fonction retourne un entier pour que l'on puisse détecter quand l'utilisateur quitte la fenêtre et donc ne pas démarrer le jeu dans la fonction main.

3.2.2 Le menu d'aide

Cette fonction est appelée quand l'utilisateur appuis sur la touche **escape** pendant le jeu et affiche un texte qui donne les différents contrôles clavier au joueur.

3.2.3 La demande de sauvegarde

Cette fonction est appelée quand l'utilisateur quitte la fenêtre (en appuyant sur **q**) sans avoir fini le jeu. Elle affiche un texte expliquant que l'utilisateur peut appuyer sur **y** pour sauvegarder sa partie, toute autre touche fermera le programme sans sauvegarde. La boucle attendant la réponse de l'utilisateur se trouve dans la fonction.

3.2.4 L'affichage de fin

Cette fonction est appelée quand l'utilisateur a complété le défi et a appuyé sur **t** ou **q**. Elle affiche "victoire" et aussi le nombre de coup que le jour a mis pour terminer le jeu à l'aide **SDL_TTF**. Elle fait appel à la fonction **attend_touche** qui permet d'attendre que le joueur appuis sur une touche pour fermer le programme.

4 Les contrôles claviers

Il y a deux fonctions qui permettent de détecter que l'utilisateur appuis sur des touches du clavier. La fonction attend touche est située dans affichage.h ne sert que à attendre que l'utilisateur appuis sur une touche et retourne 0 si l'utilisateur ferme la fenêtre.

L'autre fonction est la fonction de récupération des touche principal et intervient pendant le jeu. Elle permet à l'utilisateur de déplacer les pièces en utilisant les flèches (ou les touches de déplacement de l'éditeur vi), mais aussi de faire apparaître le menu en appuyant sur **escape**, de quitter le programme en appuyant sur **q** ou en fermant la fenêtre, d'activer l'affichage du numéro des pièces ou l'espacement entre les pièces et enfin de tester si la grille est valide. Cette fonction retourne aussi 0 quand l'utilisateur quitte la fenêtre.