SAé 1.1 DEV

Taquin

Projet réalisé par Kayyissa Haïssous et Claire Gobert
Etudiantes en première année de BUT informatique à l'IUT de Fontainebleau

_____Table des matières_____

_Introduction :

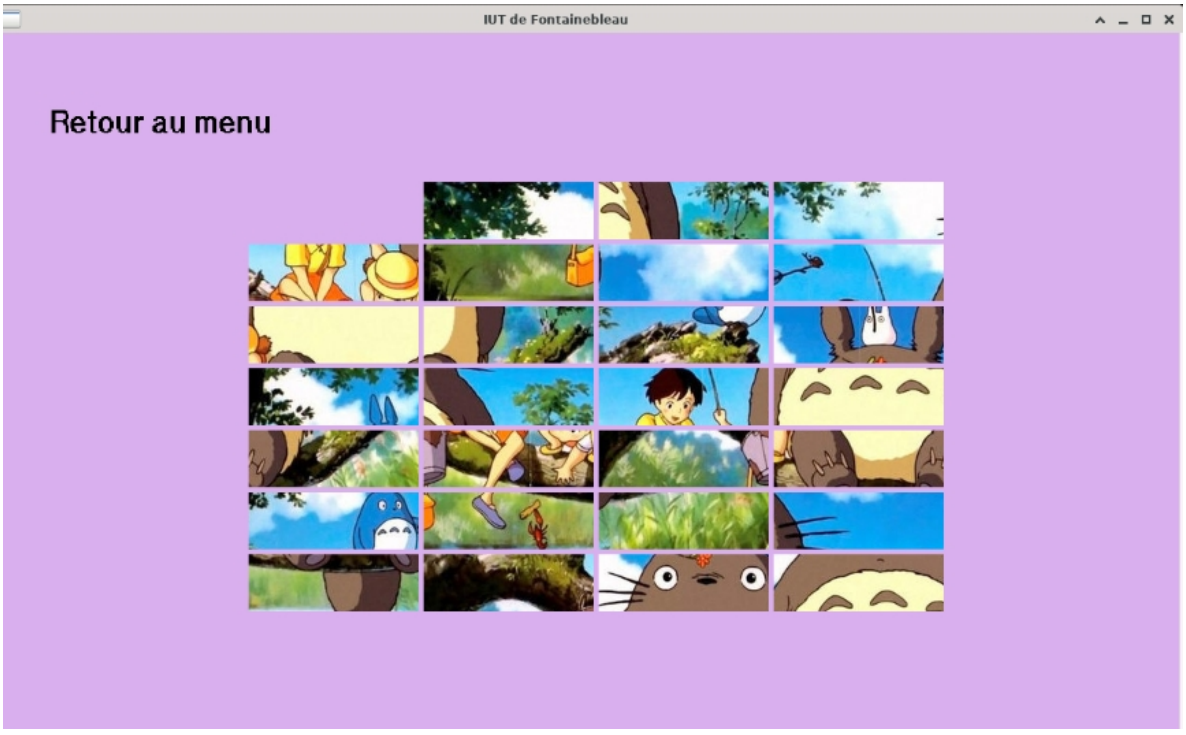
Le jeu du taquin que nous proposons permet au joueur de choisir indépendamment sur une page de menu, une des trois images qui lui sont proposées, le nombre de lignes et le nombre de colonnes en lesquelles il souhaite que son image soit découpée pour jouer. Il pourra ensuite jouer au clavier dans le but de reconstituer l'image en inversant l'emplacement vide avec un morceau d'image adjacent. Lorsqu'il réussit, le joueur est félicité et peut choisir de retourner au menu pour éventuellement relancer une nouvelle partie ou quitter le jeu.

_Fonctionnalités du programme :

Lorsque le programme est lancé, une fenêtre d'accueil et de menu s'ouvre :

L'utilisateur doit alors sélectionner son image, un nombre de lignes et de colonnes en lesquelles son image sera découpée et ses choix seront encadrés pour être visibles :

L'utilisateur a également le choix entre lancer une partie ou quitter le jeu. Soit à l'aide des boutons en bas de l'écran soit en appuyant respectivement sur <Entrer> ou sur <échap>. Lorsque le joueur lance la partie, le programme vérifie que les paramètres de jeu sont bien sélectionnés et lance le jeu :



Le joueur peut alors, à l'aide des flèches directionnelles du clavier, déplacer la case vide. Le but est de reconstituer l'image d'origine, avec la case vide située en haut à gauche :

Insérer image reconstituée

Le joueur peut cliquer à n'importe quel moment sur le bouton de retour au menu, s'il souhaite commencer une nouvelle partie ou quitter le jeu.

_Structure du programme :

Fichier Makefile :
Permet une compilation plus simple du programme.

Fichier fonctions :
Fichier d'en-tête contenant les déclarations des fonctions appelées entre plusieurs fichiers.

Fichier main :
Fichier contenant la structure principale du code et fait appel aux fonctions présentes dans les autres fichiers.
Contient également la fonction du menu.

Fonction fermer_jeu :

Termine le programme.

Fonction main :

Lance la fonction de la fenêtre du menu et contient des déclarations utiles au bon fonctionnement du programme (comme la taille de la fenêtre, les chemins vers les images…).

Fonction menu :

Ouvre la fenêtre de menu.
L'utilisateur doit régler trois paramètres à la souris :

- Choisir l'image avec laquelle il va jouer (parmi 3) ;
- Choisir en combien de ligne elle sera divisée (entre 3 et 8) ;
- Choisir en combien de colonnes elle sera divisée (entre 3 et 8).

Les paramètres choisis sont encadrés afin de permettre au joueur de les visualiser.
Deux boutons fonctionnels :

- « Lancer la partie » qui lance la fonction game si les paramètres sont choisis ;
- « Quitter » qui permet de tout quitter.

Touches fonctionnelles :

- <Entrer> doit permettre de lancer la fonction de jeu si les paramètres sont choisis ;
- <Esc> doit permettre de tout quitter.

Fichier Jeu :

Contient les fonctions relatives au jeu en lui-même.

Fonction game :

Prend également comme arguments les paramètres de jeu.
Divise l'image choisie en tuiles et les mélange :

- Divise la hauteur de l'image par le nombre de lignes (=hauteur) ;
- Divise la longueur de l'image par le nombre de colonnes (=longueur) ;
- Enregistre le nombre X d'images qu'il y aura à générer en multipliant le nombre de colonnes avec le nombre de lignes ;
- Créer X images numérotées ayant pour résolution hauteur x longueur ;
- Crée une grille de X cases ;
- appelle la fonction de remplissage ;
- appelle la fonction mélange ;
- Appelle la fonction affichage ;

Doit réagir aux flèches directionnelles :

- Si flèche du haut, vérifie qu'il existe une case i+nbcolonnes avec i la case vide, si oui place l'image x de la case i+nbcolonnes sur i ;
- Si flèche du bas, vérifie qu'il existe une case i-nbcolonnes avec i la case vide, si oui place l'image x de la case i-nbcolonnes sur i ;
- Si flèche de droite, vérifie qu'il existe une case i-1 avec i la case vide, si oui place l'image x de la case i-1 sur i ;
- Si flèche de gauche, vérifie qu'il existe une case i+1 avec i la case vide, si oui place l'image x de la case i+1 sur i.

Appelle les fonctions verifzero, affichage et fvictoire et augmente de 1 le compteur de coups joués à chaque déplacement.
Si les conditions de victoire sont vérifiées, permet de sortir de la boucle de jeu et lance la fonction victoire.

Fonction verifzero :
Détermine quel est l'indice de la case vide et le renvoie.

Fonction affichage :
Efface l'affichage actuel et affiche de nouveau les tuiles avec le nouvel ordre.
Affiche également un bouton « retour au menu » qui relance la fonction menu.

Fonction remplissage :
Remplit les tableaux grille et tuiles.

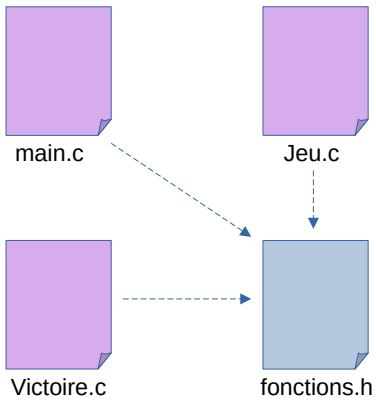
Fonction melange :
Effectue une suite aléatoire de mouvements possibles avec les tuiles afin de définir leur place sur la grille.

Fichier Victoire :
Contient les fonctions relatives à la victoire.

Fonction fvictoire :
Vérifie si les numéros de cases de la grilles correspondent aux numéros d'images.

Fonction victoire :
Félicite le joueur et permet de relancer une partie ou de quitter le jeu.

Nous avons choisi ce découpage car cela permet de séparer les différentes étapes de jeu et ainsi de mieux se répartir le travail.
Voici le diagramme qui représente ce découpage :



_Explication des données :

Une partie en cours implique que le programme possède diverses données la concernant.
Il y a la variable de type *int* **img** qui représente le numéro de l'image choisie, et *char* **chemin_img** qui représente quant à elle, le nom de l'image.
Il y a également les variables *int* **lignes** et **colonnes** choisies par le joueur.
Une variable *int* **nbcases** représente le nombre de cases de jeu de la partie.
Il existe également un compteur du nombre de coups joués (*int* **nbr_deplacement**).
Il y a aussi les variables *int* **hauteur** et *int* **largeur** qui représentent la taille d'une tuile de jeu.
Les tableaux *int** **grille** et **tuiles** dépendent aussi du nombre de cases.
Int **verifligne,** *int* **verifzero,** *int* **victoire** et bien d'autres sont variables au cours de la partie et servent à son bon déroulement.

_Garantie de la possibilité de reconstitution :

Dans la fonction « melange » le programme effectue une suite aléatoire de mouvements possibles au sein de la grille de taquin. Pour cela il choisi au hasard de déplacer la case vide (qui correspond à celle d'indice 0, soit celle en haut à gauche) vers la droite, la gauche, le haut ou le bas. Il vérifie ensuite que ce déplacement est possible.
Soit i la case vide :
Si i+1 est divisible par le nombre de colonnes (avec aucun reste), la case vide se trouve déjà tout à droite et déplacer la case i à droite est impossible ;
Si i est divisible par le nombre de colonnes (avec aucun reste), la case vide se trouve déjà tout à gauche et déplacer la case i à gauche est impossible ;
Si i est plus petit que le nombre de colonnes, la case vide se trouve déjà tout en haut et déplacer la case i en haut est impossible ;
Si i est plus grand que le nombre de colonnes multiplié par le nombre de lignes, moins le nombre de colonnes, la case vide se trouve déjà tout en bas et déplacer la case i en bas est impossible.
Ex :

0	1	i-5	3	4
5	i-1	i	i+1	9
10	11	i+5	13	14

Cette manière de mélanger garanti que l'image est reconstituable par ces mêmes déplacements.

_Conclusions personnelles

_Kayyissa :

Principalement en charge de la partie logique (remplissage, mélange, affichage, déplacement des tuiles, condition de victoire et organisation du code).
« Ce projet a prit bien plus de temps que je ne le pensais (*virée deux fois de l'IUT car il était presque 19h*). J'ai pu beaucoup apprendre et ai été forcé à beaucoup réfléchir (chose que je ne fait pas assez visiblement). Je regrette que les déplacements ne soient pas réalisables à la souris, mais je suis ravie de rendre un programme fonctionnel (je n'y croyais plus). J'espère que ce projet me servira de bonne expérience pour ne pas commettre de nouveau certaines erreurs dans le futur. Je tiens aussi à dire que je trouve que le travail d'équipe au sein de notre binôme a très bien fonctionné. (ps : ne me parlez plus jamais de taquin de toute la vie) »

_Claire :

Principalement en charge de la partie graphique (main, menu, game, victoire).
« Ce projet était très intéressant, malgré le fait que nous n'avons pas pu finir toutes les fonctionnalités. Heureusement que j'ai déjà eu l'occasion de travailler avec cette bibliothèque graphique l'an dernier (merci Denis pour ce bijou). Nous nous excusons pour l'absence d'image concernant la victoire, la réalité et que nous sommes toutes deux trop mauvaises à ce jeu même pour résoudre un taquin 3x3…»