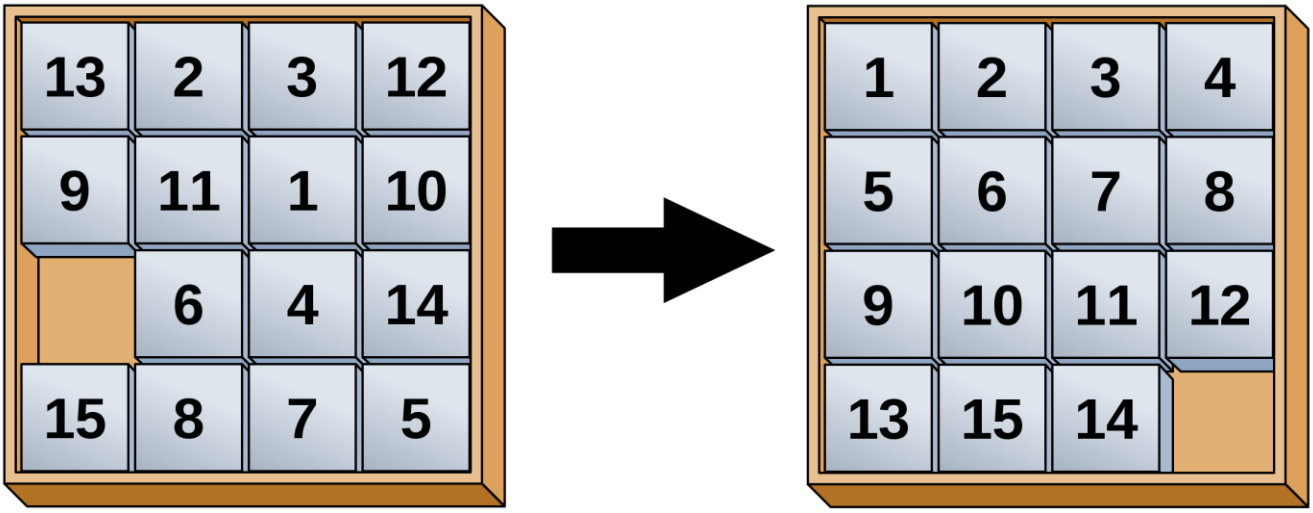
PROJET TUTORÉ

GAUDRU Rémy
&
Jordan DEMARTIN

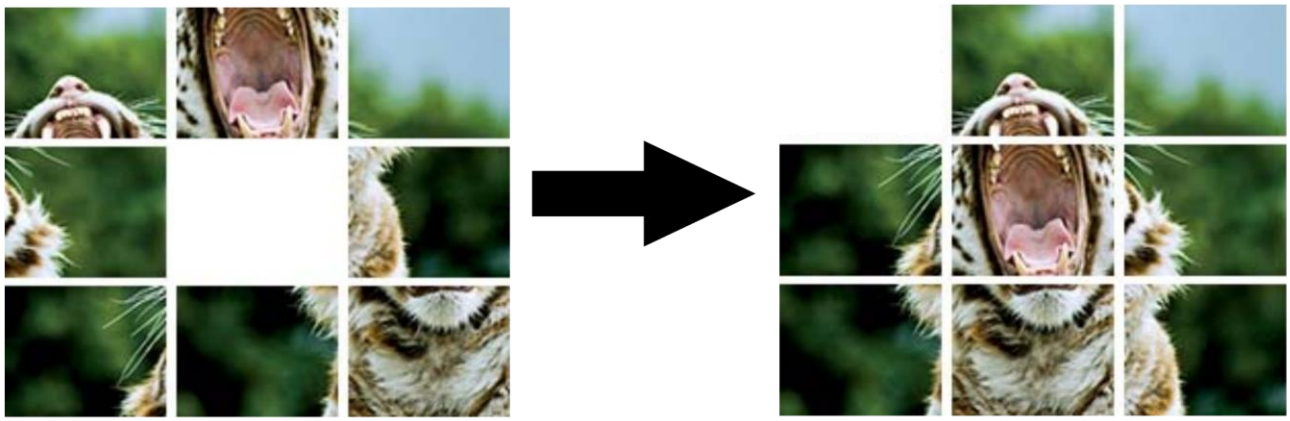
SOMMAIRE

I – INTRODUCTION

Ce projet tutoré, lancé le mardi 22 novembre 2017, consiste à réaliser en langage C (au standard 89) un petit jeu appelé Taquin.



Celui-ci consiste à remettre en ordre une série de nombre dans une grille.
Cependant pour ce sujet, nous devions non pas utiliser une suite de numéros mais une image découpée en plusieurs parties.



L'utilisateur doit pouvoir choisir son image (au moins 3 choix différents) ainsi que le nombre de lignes et de colonnes que son image possèdera.
De plus, il doit pouvoir jouer au jeu à la souris ET/OU au clavier.

Les seules bibliothèques disponibles pour réaliser ce projet sont la bibliothèque graphique de l'IUT de Fontainebleau et la bibliothèque standard du C.

II - DESCRIPTIONS DES FONCTIONNALITÉS

a) Menu

Lorsque le programme est lancé, la fenêtre ci-contre se lance.
C'est l'accueil du jeu, celui-ci est constitué de 2 boutons : JOUER et QUITTER.
Le bouton QUITTER permet de fermer la fenêtre et quitter le programme.
Le bouton JOUER, quant à lui permet d'emmener le joueur vers un second menu qui lui permettra de faire les choix qu'il souhaite avant de lancer la partie.

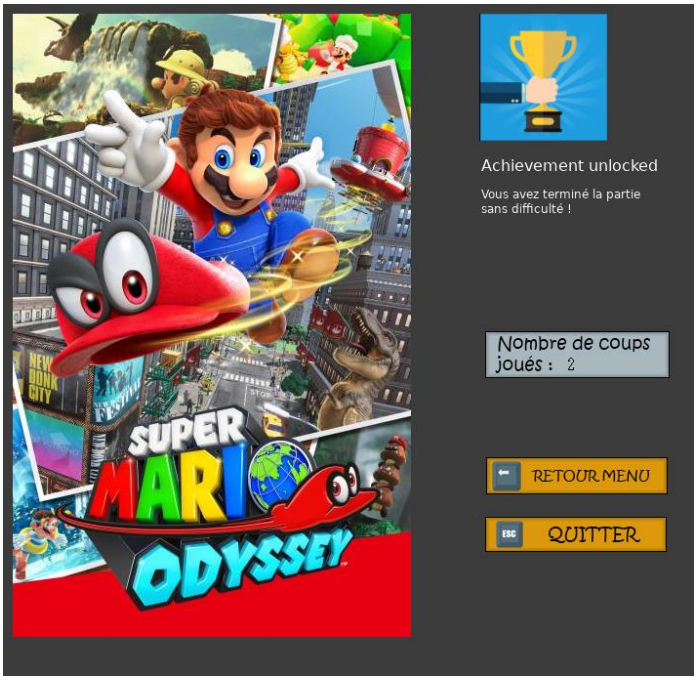
Pour choisir ses paramètres le joueur peut utiliser la souris pour cliquer sur les éléments qu'il souhaite choisir, ou bien utiliser le clavier en se repérant grâce au petit rond blanc dessiner au-dessus à gauche du paramètre où il se situe. Il devra appuyer sur "Entrée" pour valider un choix.
Enfin pour lancer la partie avec les paramètre sélectionnés l'utilisateur devra appuyer sur le bouton GO! situé en bas à droite de la fenêtre.

b) La partie

Une fois bouton GO! pressé :
La partie se lance, la fenêtre précédente se ferme et une nouvelle, adapté aux caractéristiques de la partie (image, lignes, colonnes) s'ouvre, l'image est affichée et mélanger.

Le joueur est donc maintenant lancé dans la partie, celui-ci peut déplacer les cases adjacentes à la case vide avec les flèches directionnelles de son clavier ou bien avec sa souris, en cliquant sur la case qu'il veut déplacer vers l'espace vide.

On remarquera qu'on affiche bien à l'utilisateur à droite de l'écran le nombre de coups qu'il a déjà joué, ainsi que trois options :
Remélanger l'image, retourner au menu et quitter le jeu, bien entendue, on utilise ces touches soit en cliquant dessus, soit en appuyant sur la touche représentée à droite du bouton désiré.



Enfin, si le joueur parvient à compléter l'image, celle-ci sera réaffiché en entière à la place du puzzle, l'option remélanger disparaît et le jeu félicite le joueur d'avoir réussis à terminer sa partie.
Le joueur a toujours l'option de retourner au menu ou de quitter le jeu.

III - PRÉSENTATION DE LA STRUCTURE INTERNE DU PROGRAMME

Notre programme est divisé 4 fichiers sources appelés : menu.c, set_up.c, moving.c & main.c.

menu.c : Ce fichier source contient le code qui gère l'affichage et la gestion des paramètres que l'utilisateur a choisi lors du lancement du programme.

set_up.c: Ce fichier prépare la partie, il crée une fenêtre d'une certaine taille dépendant des choix de l'utilisateur. Il coupe l'image en plusieurs morceaux et l'affiche. Puis prépare un tableau (à une dimension que l'on utilise comme un tableau à 2 dimensions) avec des valeurs allant de 0 à X (dépend du choix de l'utilisateur, 0 représentant l'espace vide où l'image peut se déplacer). Ainsi, chaque morceau d'image correspond à un numéro du tableau et on peut donc simplement vérifier l'état de l'image.

moving.c: Ce fichier gère tous les déplacements durant la partie. Il permet de controler le clavier, la souris, de mélanger l'image et les valeurs du tableau au début de la partie ou lors de l'appuie sur le bouton RE-MELANGER.

main.c: Ce fichier est le cœur du programme, il appelle les fonctions des autres fichiers et lance la partie.

IV - MODELISATION DE L'ETAT DE LA PARTIE EN COURS

Les données représentant l'état de la partie en cours sont enregistrés dans un tableau nommer 'partie', chaque case est associée à un nombre, le chiffre de la dizaine représentant la ligne de la position finale attendu et celui de l'unité représentant la colonne de la position finale attendu.
(Exception : la case que l'on devrait donc appeler '11' est nommé 0 car elle représente la case vide)

Les -1 peuvent être interprétés comme des murs, c'est à dire que si on dit à la fonction deplacement_case de faire un mouvement au niveau d'un de ceux-ci, elle renvoie une erreur et ne fais rien.

a : nombre de colonnes choisit par l'utilisateur
b: nombre de lignes choisit par l'utilisateur

-1	-1	-1	-1	-1	-1
-1	0	12	…	1a	-1
-1	21	22	…	2a	-1
-1	…	…	…	…	-1
-1	b1	b2	…	ba	-1
-1	-1	-1	-1	-1	-1

V - DÉMONSTRATION MÉLANGE FONCTIONNEL

Afin d'effectuer un mélange, nous faisons appel à 2 autres fonctions : rand() et deplacement_case().

On génère un nombre aléatoire entre 0 et 4 grâce à rand(),
Chaque numéro représentant une direction :
0=haut, 1=droite, 2=bas, 3=gauche.

Une fois le numéro choisi, on appelle la fonction deplacement_case() à laquelle on donne le numéro aléatoire généré. La fonction précédemment nommée tentera de faire bouger la case vide dans la direction demander, si c'est impossible (un mur (-1) dans la direction demander) le mouvement n'aura pas lieu.

La fonction mélange répètera 10 000 fois le processus expliqué ci-dessus.

Ainsi, la fonction mélange agit seulement dans les règles du jeu.
Prenons l'exemple du fait de devoir remettre en place le jeu alors qu'il n'y a eu qu'un déplacement vers la droite, il suffira simplement de faire un déplacement vers la gauche.

En suivant ce même principe, après 10 000 coups aléatoire joués par la fonction rand(), on pourrait effectuer la combinaison inverse de mouvements effectués précédemment pour résoudre une partie de notre Taquin.

(Devoir faire 10 000 coups pour résoudre ce puzzle est bien entendue fort peu probable car il y'a beaucoup de chance qu'il existe une combinaison de mouvement qui permette une victoire bien plus rapide.)

VI - CONCLUSION PERSONNELLE

Rémy Gaudru :

Tout d'abord, j'ai trouvé que le niveau du sujet de ce projet tutoré était très bien choisi pour une première expérience. En effet, malgré des difficultés sur certains points, tous les outils nécessaires pour les résoudre était à notre disposition.
De plus, avec mon camarade, nous avons su nous organiser afin de se répartir les tâches efficacement. Ainsi, complémentaire l'un à l'autre, le développement du projet a été rapide et efficace.
Enfin, cette réalisation m'a réellement donné l'envie de recommencer un nouveau projet (personnel) dans le but de pouvoir découvrir une nouvelle bibliothèque graphique, plus complète, afin de pouvoir de nouveau aboutir à un programme fonctionnel réalisé de A à Z par moi-même.
Pour conclure, j'ai adoré réaliser ce jeu du taquin avec mon camarade car, malgré les appréhensions, nous avons réussi à le finir et obtenir un résultat satisfaisant.

Jordan Demartin :

J'ai personnellement beaucoup apprécié travailler sur ce projet, surtout vers la moitié de celui-ci, quand nous avons travaillé sur la fonction de déplacement.
En effet, c'était très plaisant de voir après beaucoup de tests et de corrections nos idées sur le déplacement des cases se matérialiser sous nos yeux.
C'est pour retrouver une sensation de ce genre que je suis assez enthousiaste à l'idée de personnellement travailler à nouveau sur un projet du même type.

Tout cela pour dire que j'écris aujourd'hui cette conclusion en étant très satisfait de notre projet, je pense que malgré notre faible expérience nous avons réussis à produire un travail de plutôt bonne qualité.