

SAÉ 1.1 DEV : TAQUIN



Projet réalisé par :
Benamar Zemmouri et Félix Brinet

Table des matières :

- **Introduction (page 3)**
- **Fonctionnalité du programme (page 3-5)**
- **Structure du programme (page 6)**
- **Explication des données (page 7)**
- **Fonctionnement mélange (page 8)**
- **Conclusion (page 9)**

Introduction générale :

Pour jouer à notre taquin, l'utilisateur doit d'abord choisir parmi 3 trois images proposé dans le premier menu. Une fois l'image sélectionné, il peut choisir comment découper son image. L'utilisateur peut découper l'image entre trois et huit lignes et entre trois et huit colonnes. Une fois la sélection faite, l'image complète sera affichée à gauche et l'image découpé en carré mélangé avec un carré blanc sera affichée à droite. L'utilisateur pourra ensuite résoudre le taquin à l'aide de la souris ou bien du clavier. Une fois le taquin résolu, une image de victoire sera affichée et cela proposera à l'utilisateur de revenir au menu avec un bouton menu situé en haut à gauche afin de sélectionner une autre image ou rejouer. L'utilisateur pourra sinon quitter le jeu avec bouton exit en haut à droite.

Fonctionnalité du programme :

Une fois le programme lancé avec la commande make run la page du menu suivante apparaît à l'écran :



L'utilisateur doit alors choisir entre trois images.

Ensuite la page des réglages s'ouvre et l'utilisateur peut découper l'image selon le nombre de lignes et de colonnes qu'il souhaite :



L'utilisateur peut également quitter le programme en cliquant sur exit en haut à gauche.

Une fois les lignes et les colonnes sélectionné, une nouvelle fenêtre s'ouvre qui affiche l'image sélectionné à gauche et l'image sous forme de quadrillage mélangé à droite avec une case en moins :

L'utilisateur peut résoudre le taquin et cela affiche le nombre de coup qu'il a effectué en bas au milieu.



Une fois le taquin résolu, cela affiche une nouvelle page qui indique à l'utilisateur qu'il a réussi à résoudre le taquin. Il peut ensuite retourner au « Menu » pour revenir au choix des images ou bien quitter le jeu en cliquant sur le bouton « exit ».



Structure du programme :

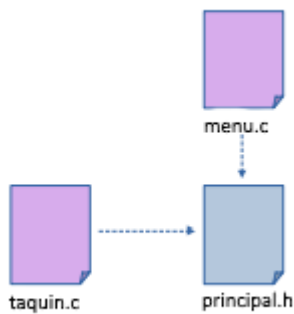


diagramme fichiers sources

Fichier Makefile : permet la compilation des différents fichiers du programme

Fonctions de jeux dans fichier principal :

- Fonction Déplacements : Il y a quatre fonctions de déplacements, une pour déplacer une tuile vers le haut, le bas, la gauche et la droite. Chaque fonction vérifie avant de déplacer à l'endroit où elle le souhaite si la case est vide.
- Actualisation du nombre de coup et de l'affichage du taquin
- Découper l'image en fonction des lignes et des colonnes prédéfinies
- Initialiser : initialise le taquin et change les tuiles de manière aléatoire et faisable.
- Fonction du Test de victoire, elle permet de vérifier si le taquin est réussi ou non. Elle renvoie 0 si le taquin est réussi.

Fichier menus : contient les différentes images, les tailles et la récupération de la position et du clic de la souris.

- Fonction menu1 : Cette fonction permet d'afficher le fond du menu de choix d'image et de récupérer quelle image est choisie par l'utilisateur à l'aide de la souris.
- Fonctions menu2 : Cette fonction permet d'afficher l'image de fond du menu de réglage et de récupérer les paramètres de découpage que l'utilisateur décide avec la souris.

- Fonction menu3 : Cette fonction permet d'afficher l'image de fond du menu de victoire. Ainsi que de laisser l'utilisateur quitter le menu avec le bouton exit ou de revenir au menu1.
- Fonctions dimension bx et by : Permet de récupérer les tailles des images.

Fichier taquin : affichage graphique du taquin (nombre de coup, image de gauche et case retirée).

- Fonction AffichageG : affichage de la fenêtre de jeu du taquin

Explication des données :

Lors de la partie du taquin, nous utilisons au début les dimensions de l'image choisie pour créer une fenêtre suffisamment grande.

Pour découper le taquin en tuiles de bonnes tailles nous avons eu besoin des coordonnées de départ de l'image(departx et departy) pour la décaler et ensuite créer les dimensions de chaque case.

Pour effectuer un clic qui soit bien valide nous vérifions que celui-ci est bien dans l'image du taquin, donc nous utilisons les coordonnées de notre souris.

Dans les valeurs qui changent durant le déroulement d'une partie, nous avons d'abord le nombre de coups qui varie à chaque coup effectué et qui nous permet ensuite de l'afficher pour que le joueur sache son nombre de coups.

Les variable i_candidat et j_candidat nous permettent de vérifier que le clic effectué par la souris est bien une case à côté de la case vide et rendant le mouvement possible. Pour vérifier cela nous avons aussi eu besoin de li_trou et col_trou.

Fonctionnement mélange :

Comme nous n'utilisons que des vraies valeurs à l'aide des lignes et colonnes du taquin, le taquin est forcément réalisable puisqu'il n'y a qu'à faire le chemin inverse pour remettre les pièces dans le bon ordre.

Pour mélanger nous prenons un nombre au hasard avec la fonction `rand()` en fonction du temps, puis nous faisons la division euclidienne par 4. En fonction du reste, la tuile est soit placée à gauche, à droite, en bas ou en haut. Cela permet de faire un mélange purement aléatoire et grâce à ceci notre taquin ne sera jamais le même puisque le nombre utilisé ne sera jamais le même.

Conclusion :

Conclusion personnelle de Félix Brinet :

Ce projet m'a permis d'améliorer mes compétences dans la programmation en langage C et rattraper mes lacunes dans certaines fonctions et notamment l'organisation du code et la compilation de plusieurs programmes avec la fonction `make`. J'ai pu aussi découvrir et apprendre l'utilisation de la bibliothèque de l'iut avec ces fonctions d'affichages graphiques diverses.

Ce projet était dur à cause du temps un peu court pour le réaliser mais surtout de la complexité d'utiliser des variables globales et une nouvelle bibliothèque.

Conclusion personnelle de Benamar Zemmouri :

Ce projet m'a permis de mieux gérer les fonctions et l'organisation du code. Cependant je n'ai pas su retirer les variables globales et je n'ai pas su maîtriser le fonctionnement des fonctions en rapport avec les touches.

Même si ce projet était à mon goût difficile, j'ai pu expérimenter la bibliothèque graphique de l'IUT et faire quelque chose qui était totalement nouveau pour moi.