

Rapport de projet : Jeu de Taquin

Tajeddine et Ibrahim

Introduction :

Ce projet consiste à développer une version du célèbre jeu de Taquin.

Le Taquin est un jeu de puzzle qui se joue sur un damier où l'on doit remettre dans l'ordre des carreaux numérotés en les faisant glisser vers une case vide. Dans notre version, nous remplaçons les nombres par une image découpée en morceaux,

Le programme a été réalisé en langage C (C89), en utilisant exclusivement la bibliothèque graphique de l'IUT pour l'interface.

Fonctionnalités :

Voici le déroulement d'une partie en 3 étapes :

1) Menu Principale et Configuration :

Au démarrage, l'utilisateur arrive sur un menu lui proposant de choisir parmi trois images différentes.



Une fois l'image sélectionnée, le joueur définit la difficulté du puzzle en choisissant :

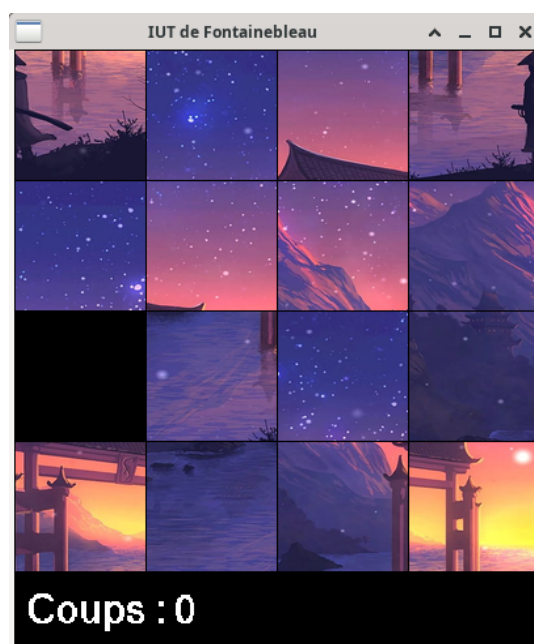
1. Le nombre de lignes (entre 3 et 8).
2. Le nombre de colonnes (entre 3 et 8)



2) Phase de Jeu :

Le plateau de jeu s'affiche avec l'image mélangée. La case vide (en noir) permet d'effectuer les mouvements.

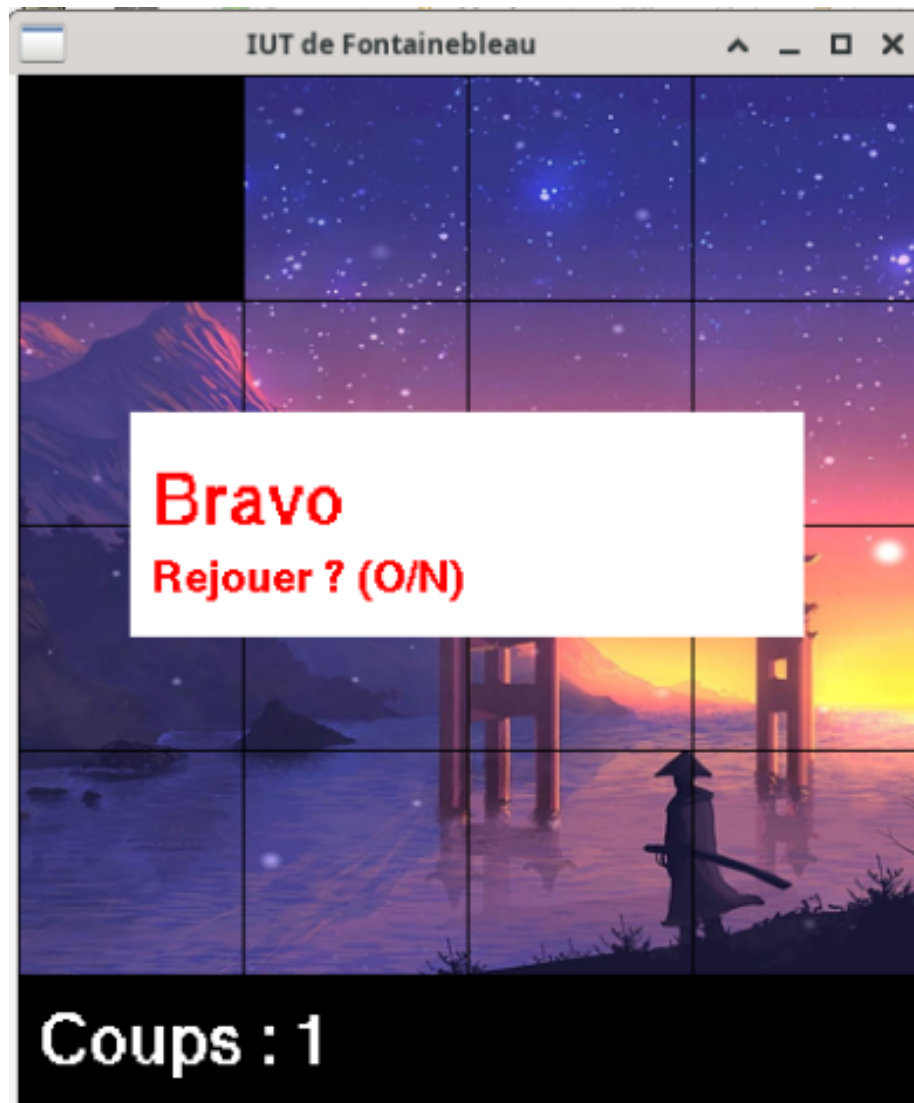
- Interaction Clavier : Les flèches directionnelles permettent de déplacer la case vide.
- Interaction Souris : Un clic sur une tuile adjacente à la case vide déplace celle-ci.
- Informations : Un compteur de coups est affiché en permanence en bas de l'écran pour suivre la progression.



3) Fin de partie :

Lorsque l'image est entièrement reconstituée, le jeu détecte automatiquement la victoire. Un message "BRAVO" s'affiche, et le joueur a le choix entre :

- Rejouer (retour au menu principal).
- Quitter l'application.

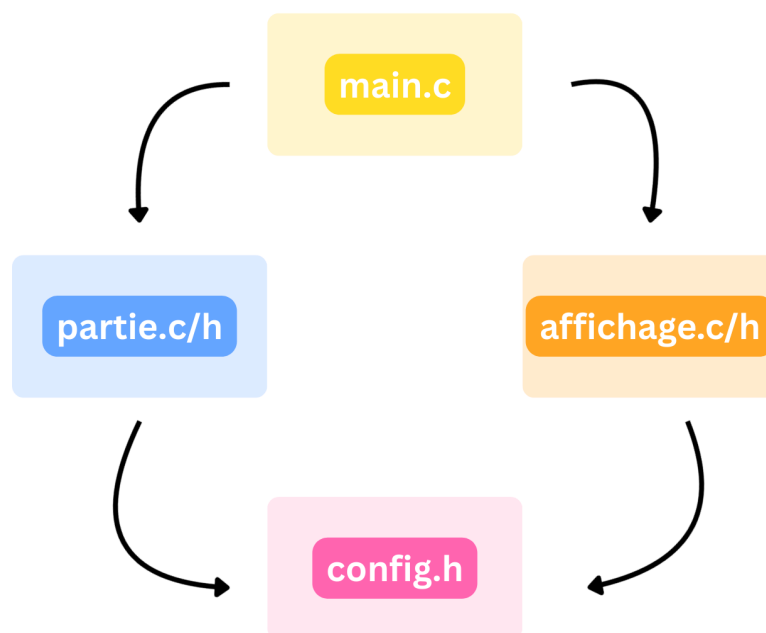


Structure du Programme :

Le projet est divisé en plusieurs fichiers :

- **config.h** : Ce fichier contient une constante globale MAX_TAILLE. Il permet de modifier les paramètres généraux du jeu sans toucher au code source.
- **partie.c / partie.h** : Ils gèrent toute la logique. Il contient la représentation de la grille , les méthodes de déplacement, de mélange et de vérification de victoire. Il est totalement indépendant de la bibliothèque graphique.
- **affichage.c / affichage.h** : ils gèrent uniquement l'interface visuelle. Il s'occupe de créer la fenêtre, charger les images, découper les sprites et dessiner la grille à l'écran.
- **main.c** : Il mène le déroulement du jeu en faisant le lien entre les entrées utilisateur (clavier/souris), la logique (**partie**) et l'affichage (**affichage**).

Ce schéma illustre comment les fichiers interagissent entre eux :



Ce diagramme montre les dépendances de compilation : le module principal (**main.c**) orchestre les modules logique (**partie**) et graphique (**affichage**). Le fichier **config.h** est placé à la base car il contient la constante partagée par l'ensemble du projet.

Représentation des Données

L'état d'une partie en cours est représenté par une structure de données simple mais efficace : un tableau à deux dimensions entières.

`int grille[MAX_TAILLE][MAX_TAILLE];`

- Type : `int` (entier).
- Signification des valeurs :
 - La valeur 0 représente la case vide (le trou).
 - Les valeurs 1, 2, 3, etc., représentent les identifiants des tuiles de l'image originale (dans l'ordre de lecture : ligne par ligne).
- Gestion de la taille : nous allouons une grille de taille maximale (8x8) définie dans `config.h`, mais nous n'utilisons que la portion définie par les variables `nb_ligne` et `nb_colonne` choisies par le joueur.

Cette représentation permet de séparer les données (les chiffres dans la grille) de leur rendu visuel (l'image affichée).

Algorithme de Mélange et Preuve de Résolubilité

Dans un Taquin, si on place les tuiles totalement au hasard, 50% des configurations obtenues sont impossibles à résoudre.

Pour contourner ce problème, nous avons utilisé une approche par simulation.

1. On part de l'état résolu (l'image est bien rangée, le vide est en haut à gauche).
2. Nous exécutons une boucle de 500 coups.
3. À chaque coups, nous choisissons aléatoirement un coup valide (Haut, Bas, Gauche ou Droite) et nous l'appliquons à la grille.

Dans le jeu de Taquin, chaque mouvement est réversible. Si l'on déplace une tuile vers la gauche, on peut la re déplacer vers la droite pour revenir à l'état précédent.

Par conséquent, il existe une suite de mouvements inverses qui permet de revenir à l'état initial de l'image.

Conclusion Personnelle

Conclusion de Ibrahim : Personnellement j'ai trouvé ce projet assez amusant ,il m'a permis de voir concrètement ce que l'on peut réaliser après déjà 3 mois de cours,ça change des exercices de TP. Hormis ça, ce projet m'a permis de consolider mes connaissances en langage C notamment avec les notions ou j'avais du mal (Organisation du code , Tableau,....).

Conclusion de Tajeddine : Personnellement j'ai trouvé ce projet très intéressant.

C'était une super expérience de développement, on a dû coder en C89, et même si c'était un peu old-school ça nous a bien forcés à bosser sur les bases : la programmation structurée, les boucles etc. L'utilisation de la bibliothèque graphique de l'IUT, c'était le vrai challenge intéressant et la partie que j'ai aimé le plus car c'était plutôt intuitif. Ce projet a été l'occasion de mettre en pratique tout ce qu'on a appris, surtout pour organiser le code de manière propre (organisation du code).