

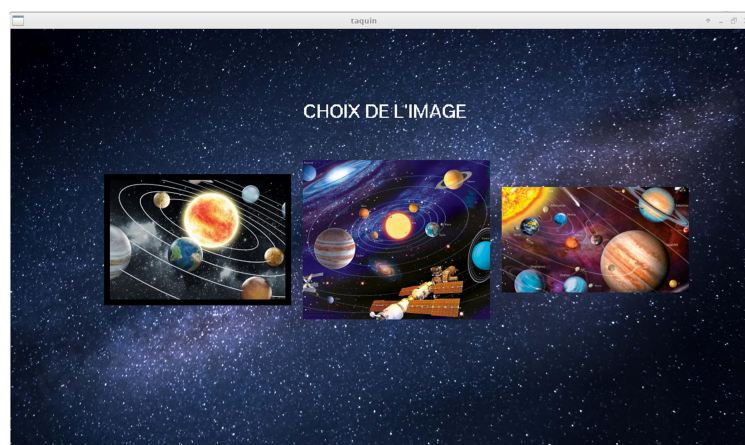
# Jeu de taquin

## Introduction :

Ce jeu consiste à déplacer des cases d'une image découpée en un certain nombre de lignes et de colonnes permettant, une fois les cases mélangées, de se déplacer et reconstituer l'image en un minimum de coups possibles.

## Fonctionnalités :

L'utilisateur a le choix entre 3 images.

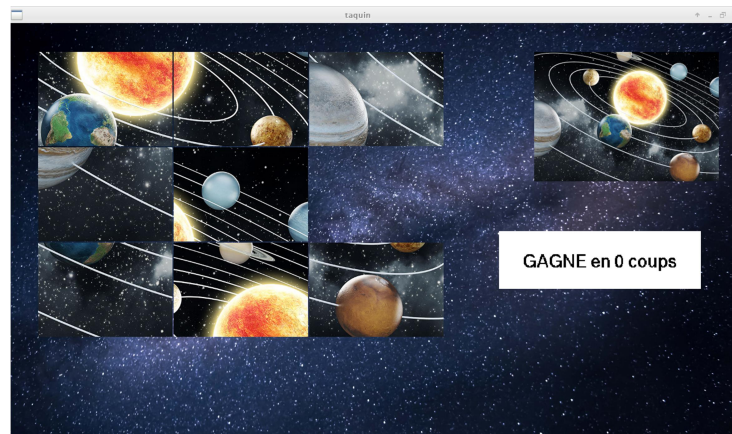


Après avoir sélectionné son image, il choisit indépendamment le nombre de lignes et de colonnes puis il valide son choix.

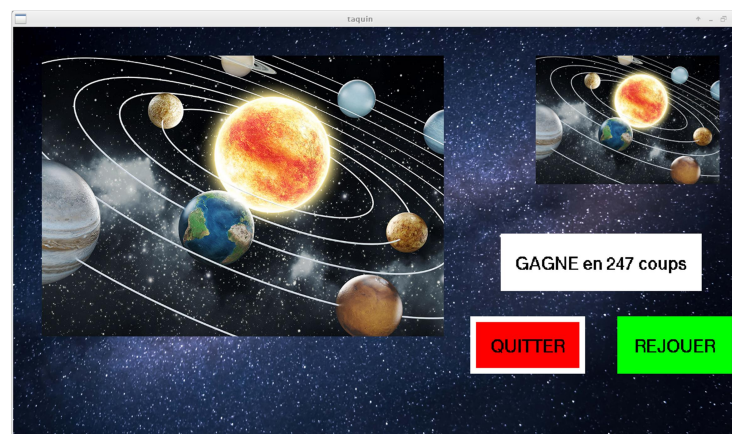


David Asselineau  
Flavien Perineau

Une fois les cases mélangées, l'utilisateur peut alors commencer à jouer. Il peut se déplacer avec le clavier (souris partiellement fonctionnelle) et ainsi remettre en ordre les cases pour gagner.



Lorsqu'il gagne, il peut ensuite recommencer une partie ou quitter le jeu.



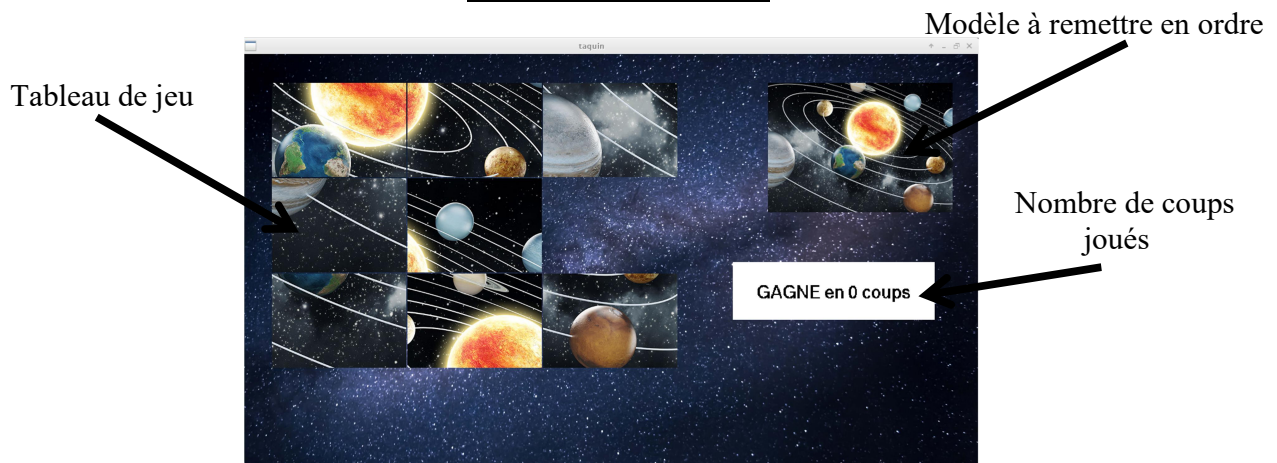
## Structure interne du programme :

Le programme est découpé en plusieurs parties :

- menu.c comprend le menu de sélection de l'image et du choix des lignes et des colonnes
- logique.c s'occupe de toute la partie logique donc du tableau de valeurs contenant le numéro des cases du jeu, du déplacement des cases du tableau au clavier et à la souris, du mélange et de l'initialisation du tableau.
- affichage.c permet l'affichage des cases du jeu selon le tableau et d'intervertir 2 cases.
- main.c rassemble tous les fichiers source et s'occupe d'initialiser et de gérer le fonctionnement du programme en appelant les fonctions des différents fichiers en fonction des conditions.

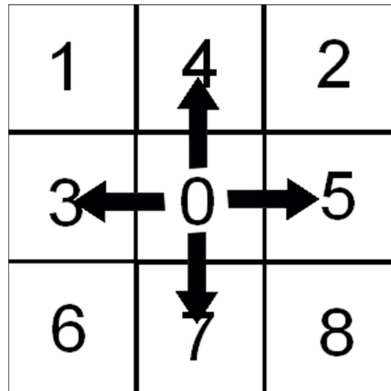
On a choisi de découper le programme en ces fichiers pour alléger le nombre de lignes qu'occuperai un seul fichier source et permettre de séparer précisément les parties principales.

## Partie en cours :



## Algorithme de mélange :

L'algorithme de mélange est construit de manière à intervertir 2 cases voisines en rendant aléatoire le fait de choisir 1 direction parmi les 4 directions possibles, et le tout 1000 fois afin de mélanger le plus de cases possibles sans que ce mélange ne prenne trop de temps.



Ce mélange permet ainsi de simuler un joueur en lui faisant appuyer sur des directions beaucoup de fois et de manière aléatoire, on est donc sûr que ce mélange permet au jeu d'être solvable.

## Conclusion :

### David :

Ce projet a été une dure épreuve car la partie logique dont je me suis occupée s'est avérée difficile à cause des inversions entre les lignes et les colonnes du tableau et de notre choix de mettre des -1 autour des numéros des cases dans le tableau pour éviter de se déplacer à des endroits non voulus. Tout cela a provoqué des problèmes lors de l'affichage des cases qui nous ont bloqués jusqu'à la fin du projet. Ce projet a quand même permis d'améliorer mes connaissances en langage C.

### Flavien :

Au début du projet, j'ai essayé de répartir le travail pour nous deux, en définissant tous les deux les entêtes des fichiers source mais à cause de la précision de mes explications et de la difficulté de gestion des tableaux à 2 dimensions (inversion des lignes et des colonnes), on a eu des problèmes à la fin du projet. Mais grâce à ce projet, j'ai pu découvrir la gestion d'interface graphique et des événements de clavier et de souris.