

Rapport de projet SAÉ 1.1 Le taquin

Introduction :

Notre première SAÉ porte sur le sujet du taquin. Nous devons en C89 et à l'aide de la bibliothèque graphique de l'IUT faire un taquin complètement fonctionnel.

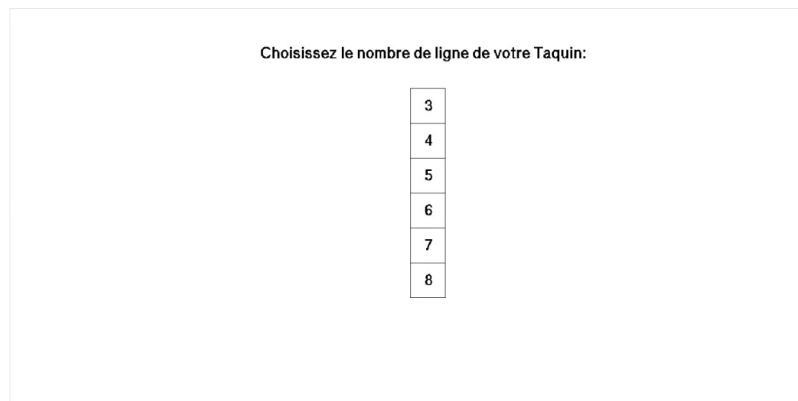
Habituellement un taquin est composée de nombres mais dans ce projet nous devons placer des images pour en construire une finale. C'est pour cela que nous devons proposer trois images avec le choix du nombre de découpage des lignes et des colonnes. Le joueur doit pouvoir choisir l'image qu'il souhaite, une fois l'image sélectionnée, elle est découpée selon les choix précédents puis la case en haut à gauche est retirée. Une contrainte nous est ajoutée nous devons proposer au joueur de choisir de entre la sélection au clavier ou à la souris. L'image est mélangée aléatoirement et le joueur doit retrouver la configuration initiale.

Description des fonctionnalités :

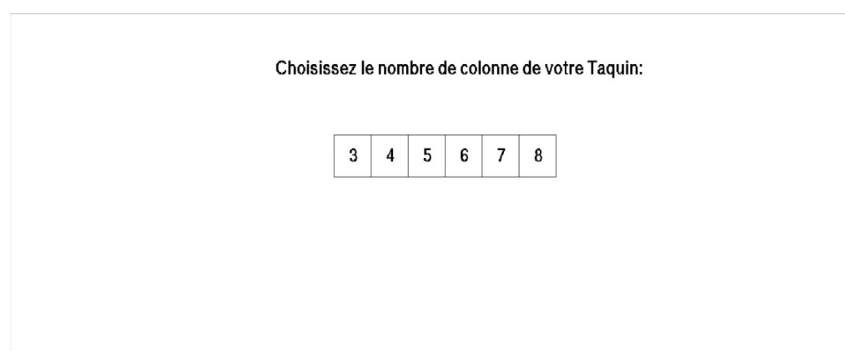
L'une des premières fonctionnalités de notre code est le choix de l'image, en effet le joueur lance notre code et voit une fenêtre avec 3 images de différentes résolutions. Il peut choisir quelle image il souhaite avoir dans son taquin à l'aide de sa souris.



Une fois l'image sélectionnée, le choix du nombre de lignes lui a été proposé.



Puis le choix du nombre de colonnes est demandé.



Le mode de déplacement.

Choisissez le mode de déplacement :

+

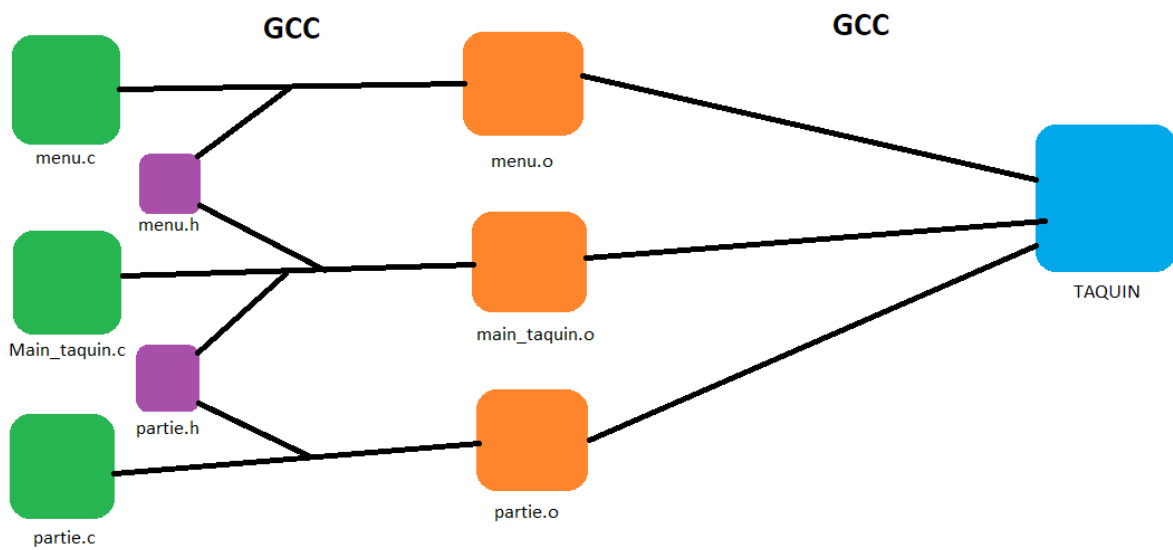
Clavier

Souris

Une fois les choix effectués l'image découpée avec la case vide apparaît.
 Puis celle-ci est mélangée aléatoirement sans aucune intervention du joueur.
 Après le mélange, le joueur peut directement commencer à résoudre son taquin.



Structure de programme :



Explication des données :

Tout d'abord, on affiche notre image puis on sauvegarde chaque coordonnée de chaque tuile dans un tableau multidimensionnel. Pour déplacer les cases on interverti chaque valeur entre elles. On créer un tableau case_vide ou de tab qui va constamment sauvegarder la position de la case vide dans notre taquin. Grâce à cela, on pourra à tout moment savoir où est notre case vide, dans le mélange aléatoire, mais aussi dans le déplacement du joueur.

Utilisation de pointeur pour renvoyer plusieurs valeurs lors de l'appel d'une fonction.

Algorithme de mélange :

Notre algorithme de mélange est assez simple. Nous avons mélangé en utilisant seulement des mouvements possibles par le joueur. Grâce à la bibliothèque `<time.h>`, on génère un nombre aléatoire entre 1 et 4 qui correspond à un déplacement, haut, bas, gauche, droite. Puis on exécute ce que la variable aléatoire à renvoyé. On peut donc être sûr qu'avec cet algorithme, le résultat sera toujours une disposition que le joueur peut reconstituer.

Conclusion personnelle :

Conclusion d'Etann De Sousa Alves :

Pour moi, ce projet représentait un vrai défi, car on nous a donné comme obligation d'utiliser une bibliothèque jusque-là encore inconnue. Donc avant de pouvoir réellement commencer à coder et réfléchir à notre plan de code, j'ai dû apprendre chaque fonction et exécuter beaucoup de codes test. Je me suis occupé de la partie déplacement avec le clavier et découpage et mélange aléatoire de l'image. J'ai passé énormément de temps à réfléchir à la bonne tactique et à la façon de découper mon code. J'ai pu progresser sur la partie gérer plusieurs fonctions, makefile et les adresses. Je n'avais personnellement jamais passé autant de temps sur un code. Pendant le projet, c'est clairement le temps qui a été la plus grosse difficulté.

Conclusion de Rayan Amoussa :

Amoussa Rayan
De Sousa Alves Etann

C'était vraiment dur de faire ce projet, j'ai eu des erreurs que je n'avais encore jamais eu, limite de mémoire, etc. Gérer un code aussi grand et apprendre la bibliothèque graphique en même temps était extrêmement dures mais aussi très formateur.

Je me suis occupé de la partie déplacement à la souris, mais aussi du menu et des fichiers .h. On devait corriger chaque bug qu'on avait et on a aussi perfectionné notre utilisation du git et du makefile ainsi des transmissions de données entre chaque fonction.