

Rapport

ROY Adrien
THEBAULT Alexandre

Introduction :

Le but de ce projet est de réaliser un jeu du taquin en langage C et de le programmer graphiquement à l'aide de la bibliothèque graphique de l'iut.

Fonctionnalités de notre programme :

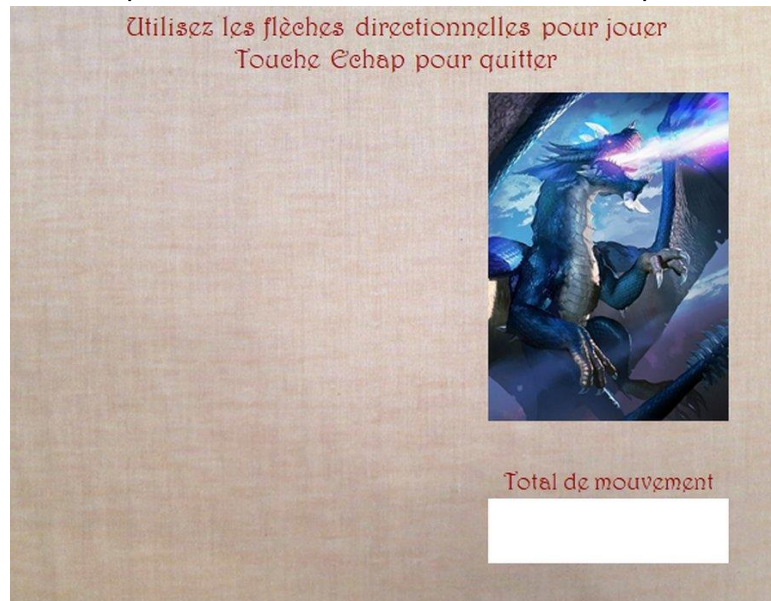
Possibilité de choisir le nombre de colonne et de ligne de la grille de jeu



Possibilité de choisir entre trois images de jeu différentes



Compteur du nombre de mouvement en temps réel



Ecran de victoire avec le nombre de coup effectués ainsi que la possibilité de rejouer



Structure interne du programme :

On a une boucle, dans laquelle il y a tout le programme (excepté la génération de la fenêtre), qui permet de retourner sur le menu lorsque l'on gagne le jeu.

Le programme dans cette boucle se déroule comme suit :

- menu principal, choix des images, du nombre de colonne et de ligne.

- remplissage des tableaux correspondants aux différents morceaux du taquin, mélange des valeurs des tableaux (donc mélange des morceaux).

- boucle qui affiche les tuiles et qui réitère à chaque mouvement. D'abord elle affiche les morceaux et trouve la position de la case vide en même temps, puis elle regarde quel mouvement on effectue (haut, bas, droite, gauche) et modifie les tableaux liés aux tuiles, et fini par vérifier si les tableaux sont dans le bon ordre (vérification de victoire). S'il ne l'est pas, la boucle recommence. S'il l'est, le menu de victoire est affiché et sort de la boucle.

- Lorsque l'on sort de la boucle précédente et que l'on ne quitte pas, la boucle principale est réitérée, et le jeu recommence à partir du menu principal.

Etat d'une partie en cours

Deux tableaux correspondent aux différentes tuiles du jeu. Il ya un tableau qui contient les coordonnées x de l'endroit où l'on coupe l'image et l'autre tableau contient les coordonnées y.

Lorsque l'utilisateur fait un déplacement, certaines valeurs dans ces tableaux vont être modifiées, en fonction du déplacement effectué.

Exemple : Le 0 correspond à la case vide

1	2	0	3	4	5	6	7	8	...
---	---	---	---	---	---	---	---	---	-----

Ce qu'il se passe c'est que la valeur correspondante à la case vide, va s'échanger avec une autre valeur, en fonction du déplacement

Exemple : mouvement vers la droite

1	0	2	3	4	5	6	7	8	...
---	---	---	---	---	---	---	---	---	-----

On a échangé la valeur précédente du 0 avec le 0, la case vide va se déplacer vers la gauche. Si l'on veut faire un mouvement vers le bas, on ne prend pas la valeur suivante (mouvement vers la droite), ni la valeur précédente (mouvement vers la gauche), mais on prend la valeur de la colonne (3 à 8) et on regarde n-fois cette valeur après la case vide.

Exemple : mouvement vers le bas, quand la colonne = 4

1	5	2	3	4	0	6	7	8	...
---	---	---	---	---	---	---	---	---	-----

Pour aller vers le haut, on n'ajoute pas mais on enlève n-fois la valeur de colonne

Garantie de la résolvabilité :

Pour mélanger notre grille de jeu, nous faisons un certain nombre de permutations (mouvements). Or si les mouvements sont faits dans un sens, ils peuvent forcément être fait dans l'autre. On peut donc conclure que notre jeu est solvable dans tous les cas.

Conclusion :

Adrien : Mêler la nouveauté de la bibliothèque graphique à la création d'un jeu grâce à toute nos connaissances acquises jusqu'à maintenant m'a beaucoup motivé. J'ai trouvé ce projet très intéressant et je suis assez satisfait du résultat ! J'espère qu'il vous a plu également !

Alexandre : Enfin un projet qui m'a vraiment donné l'impression de travailler pour quelque chose. C'est bien la première fois que je travaille avec plaisir chez moi. Le seul regret a été qu'on s'y ai mit un peu tard, mais puisque ça me faisait plaisir de construire ce jeu, j'ai travaillé intensément, en tout cas beaucoup plus que dans mes habitudes.

