

# Projet Taquin

(22/12/2017)

## Présentation du Projet :

Notre Projet consiste à créer un petit jeu nommé Taquin, dont le but est de remettre des images dans le bon ordre. Sachant qu'on peut faire glisser les images grâce à une case blanche. Pour cela, vous avez à disposition différent moyen pour les déplacer : soit avec le clavier (grâce au touche directionnelle), soit avec la souris en cliquant sur l'une des images à côté de la case blanche.

## Fonctionnalité du programme :



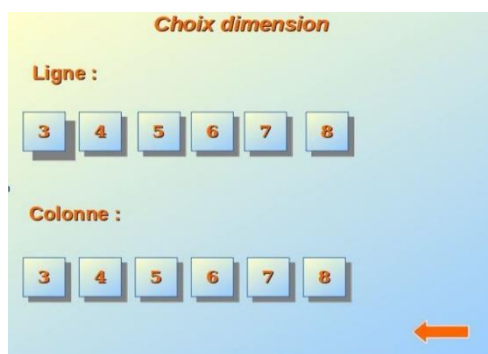
- Lors du lancement du programme vous tombez sur le menu, où deux choix-vous sont proposés : soit quitter, soit lancé une partie.



- Suite à votre choix de lancer une partie vous tombez sur les paramètres de la partie :

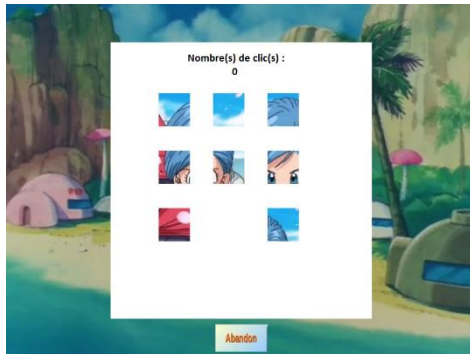
-Le choix de l'image avec laquelle vous aller jouer.

-La dimension à laquelle va être divisé l'image choisie.





- Puis vous arriverez sur la page de pré-lancement ou on vous demandera si vous êtes prêt pour lancer la partie.



- Ensuite la partie commence avec le plateau de jeu et votre score (nombre de clique)



- Pour finir la page de victoire qui montrera votre score obtenu et la possibilité de recommencer une partie.

## Structure interne du programme :

Notre Projet est divisé en 6 programmes principaux ,1 programme de lancement (celui qui contient le main), et 2 programme de calcul.

Donc il est divisé en 9 parties :

-Le fichier *main* est le programme qui contient le main et donc qui va initialiser le lancement du jeu en appelant la fonction menu.

-Le fichier *menu* qui contient la fonction qui va afficher la première interaction avec l'utilisateur en utilisant la souris ou le clavier.

-Le fichier *paramètre* est la partie de programme où s'occupera de faire le lien entre l'utilisateur et le paramétrage du jeu, c'est ici qu'il choisira son image et sa dimension pour le découpage.

-Le fichier *découpage* contient la fonction qui va calculer les coordonnées de chaque image et les mettre dans un tableau en fonction de des dimensions et de l'image choisit précédemment.

-Le fichier *mdj* contient la fonction qui va adapter les valeurs entré par l'utilisateur pour le lancement de la partie.

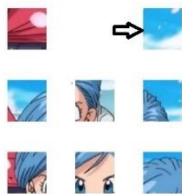
-Le fichier *plateau* contient une fonction qui va être le moteur du jeu en appelant d'autre fonctions qui sont définies dans le fichier permettant de faire une tache particulière comme afficher les images, afficher le compteur, etc.

-Le fichier *jeu* contient la fonction qui va modifier le tableau qui représente les images en fonction des actions de l'utilisateur, cette fonction est appeler par la fonction *plateau*.

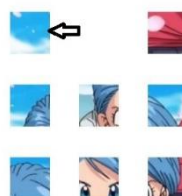
-Le fichier *générateurs* contient deux fonctions de génération différentes : l'un remplit un tableau de valeur aléatoire, l'autre il remplit un tableau et il échange de place la valeur 1 avec une valeur à côté de lui, en le faisant plusieurs fois pour bien mélanger le tableau (pour l'instant dans notre projet, nous utiliseront seulement cette fonction.

-Le fichier *victoire* contient la fonction de vérification de victoire (donc que les images sont bien remises dans le bon ordre) et la fonction d'affichage de victoire avec les options de recommencer une partie ou de quitter.

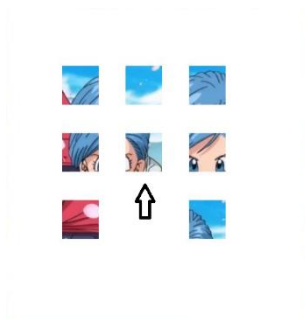
### Explication d'une partie en cours :



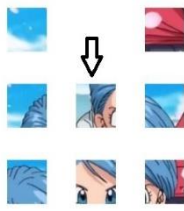
Droite : grille  $\{i, j+1\}$



Gauche grille  $\{i, j-1\}$



Haut : grille  $\{i-1, j\}$



Bas : grille  $\{i+1, j\}$

### Algorithme de mélange :

Le but de l'algorithme est de déplacer la case numéro 1 qui représente la case vide. Dans un tableau on va placer les valeurs ordonnées qui auront pour capacité la ligne \* colonne que le joueur aura donné lors de la sélection de dimension. Ensuite pour effectuer le mélange nous avons généré une valeur comprise 1 et 4 pour définir les quatre mouvements possibles (haut, bas, droite, gauche) qui vont ainsi remplacer la case 1 par la valeur de la case en mouvement et vice versa la case en mouvement devient la case 1. Ce système sera répété 100 fois pour effectuer le mélange avec sûreté. Si le joueur effectue le chemin inverse normalement il devrait retourner à l'image initiale par conséquent cela vérifie que le joueur peut reconstituer l'image.

### Conclusion Personnel :

-Pierre Villiers :

Ce projet fut agréable du point de vue de la programmation car il y avait plusieurs difficultés qui étaient présentes comme : faire un travail à plusieurs (répartir les tâches de travail, se coordonner avec mon camarade sur l'utilisation de certaines fonctions, etc...), trouver et utiliser correctement les bibliothèques qui sont nécessaires pour la construction du programme, la gestion du temps de travail et les contraintes particulières du projet. Mais ces difficultés m'ont fait apprendre de regarder un programme de différents points de vue pour trouver la solution.

-Renault Steven :

Ce projet a été très instructif, j'ai pu enrichir mes connaissances et améliorer mon travail d'équipe avec mon camarade. Même si nous avons rencontré quelques difficultés à fusionner les différents programmes que chacun de nous avons faits de notre côté, nous avons pu résoudre nos problèmes et avoir le rendu final que nous voulions.