

Jeu de taquin

I. Préambule

	1	2	3	10	6	4	12
4	5	6	7	1	14	3	7
8	9	10	11	5	15	11	13
12	13	14	15	8		2	9

Figure 1: Notre position idéale et une position engendrée aléatoirement

Le jeu de taquin (Fifteen Puzzle) est constitué d'un plateau carré, subdivisé en seize cases, sur lequel on pose quinze pièces carrées numérotées de un à quinze. Il reste donc une case libre ou trou. Le seul mouvement ou coup autorisé consiste à faire glisser l'une des pièces adjacentes au trou vers celui-ci, ce qui revient à échanger leurs positions respectives.

Ce jeu, imaginé par Sam Loyd dans les années 1870, suscita immédiatement un grand intérêt à travers l'Occident. L'une des raisons peut-être de ce rapide succès était une récompense de mille dollars promise par Loyd à quiconque parviendrait à transformer une position de départ en une position d'arrivée fixée.

De façon relativement abstraite, on peut voir l'espace des configurations du jeu de taquin comme un graphe où chaque sommet représente une configuration et où les successeurs d'un sommet sont les configurations obtenues grâce à un coup quelconque. Chaque sommet a donc deux, trois ou quatre successeurs. Le jeu consiste alors à trouver un chemin à travers ce graphe entre un sommet de départ et un sommet d'arrivée fixés.

Loyd était taquin: les positions de départ et d'arrivée qu'il avait fixées ne sont en fait reliées par aucun chemin dans le graphe des configurations. Ce fait a été publié dès 1879, et se démontre aujourd'hui facilement. Dans le cadre de ce projet, la position de départ sera variable, et la position d'arrivée sera celle illustrée à gauche de la figure 1.

II. Le sujet

Réaliser un programme en java calculant de différentes manières une solution d'un jeu de taquin. Votre programme permettra:

- de lire une position initiale,
- de tester si le problème est soluble,
- de jouer en mode interactif sur un terminal,
- de calculer une solution et de la présenter sous la forme d'une animation,
- de comparer et d'évaluer les différentes méthodes proposées.

Pour candidater à note supérieur 16/20, votre système devra traiter des problèmes de jeu de taquin de formes non nécessairement rectangulaires.

III. Calcul d'une solution (A venir)

IV. Que doit faire votre programme ?

Le programme java sera composé d'une unique archive jar **taquin.jar**. Ce programme devra s'exécuter dans un terminal avec les options suivantes :

- o `java -jar taquin.jar -name` affiche vos noms et prénoms
- o `java -jar taquin.jar -h` rappelle la liste des options du programme
- o `java -jar taquin.jar -sol fichier.taq -j` test si le jeu a une solution à partir de la position initiale donnée dans le fichier `fichier.taq`. Le format d'un fichier `taq` est le suivant : largeur, longueur, liste des valeurs contenues dans le tableau avec 0 représentant la case vide.
- o `java -jar taquin.jar -joue fichier.taq` permet à l'utilisateur de jouer sur le terminal. Le fichier `fichier.taq` contient la position initiale du jeu.
- o `java -jar taquin.jar -cal delai algo fichier.taq` calcule une solution en utilisant l'algo `algo`. Votre programme renvoie une solution (si possible) sous la forme d'une liste de positions (une par ligne). Si le temps d'exécution de votre programme excède la durée `delai`, il doit s'interrompre brutalement.
- o `java -jar taquin.jar -anime delai algo fichier.taq` est identique à la commande précédente sauf que la solution est présentée sous la forme d'une animation.
- o `java -jar taquin.jar -stat delai algo fichier.taq` est identique à la commande précédente sauf que le programme renvoie des statistiques sur l'exécution du programme : taille de la solution, nombre de positions parcourues et temps d'exécution.
- o `java -jar taquin.jar -stat delai fichier.taq` applique la commande précédente à tous les algos et renvoie sous la forme d'un tableau html toutes les statistiques.
- o `java -jar taquin.jar -aleatoire n largeur hauteur delai fichier.taq` applique tous vos algorithmes à `n` positions initiales générées de façon aléatoire de taille `largeur` x `hauteur`. Vous donnerez pour chaque algorithme : le nombre de problème résolu, et les moyennes des tailles des solutions, du nombre de solutions parcourus et du temps d'exécution. Les résultats seront présentés sous la forme d'un tableau html.

La liste des options `algo` sera donnée dans un avenir proche .

Pour candidater à une note supérieur à 16/20, les commandes précédentes devront s'appliquer à des formes non nécessairement rectangulaires. Pour les distinguer des commandes précédentes vous ajouterez l'option `-plus`.

V. Quoi et quand rendre ?

« Sera précisé en cours »

VI. Quelles références disponibles sur la toile et à la bibliothèque des sciences

« A venir »