

## Le jeu de Taquin

Le jeu de taquin se présente comme un cadre carré dans lequel coulisent des petites cases (en bois, en plastique ...). Initialement, ces cases ne représentent rien, sauf le désordre. Il est possible de les déplacer grâce à un vide ménagé par une case libre.

3	2	5
4	1	
7	8	6

À chaque étape, on fait glisser l'une des cases adjacentes à la case libre pour faire évoluer la configuration. Le casse-tête est résolu quand toutes les cases sont ordonnées et que la case libre se trouve en dernière position en bas à droite. C'est ce que nous appelons la *configuration gagnante*.

1	2	3
4	5	6
7	8	

### Mise en œuvre

L'objet du TP est de construire un paquetage `Paq_taquin` permettant de mener une partie de taquin. À la fin du TP, vous rendrez une disquette contenant un répertoire `Taquin`, et dans ce répertoire deux fichiers : `Paq_taquin.ads` et `Paq_taquin.adb` (et ces deux fichiers seulement).

Pour que le paquetage soit le plus souple possible, l'utilisateur doit avoir le choix de la taille de la grille du jeu (qui est toujours carrée, toutefois) et du type des éléments dans les cases (tant que ce type est muni d'une relation d'ordre). La configuration d'un jeu de taquin est ainsi caractérisée par

- un paramètre  $p$ , pour la longueur de la grille,
- une grille de  $p^2$  cases, constituée d'éléments d'un type générique discret `Element`,
- deux coordonnés  $x$  et  $y$ , indiquant la position de la case libre dans la grille.

**Question 1.** Écrire la spécification du paquetage générique `Paq_taquin`, avec la définition du type `Taquin`. Vous ajouterez les primitives au fur et à mesure des questions à venir.

**Question 2.** Écrire la procédure `Initialisation(T:in out Taquin)`, qui met le jeu de taquin dans une configuration initiale aléatoire. La case libre se trouve en  $(p, p)$ , en bas à droite. Les autres cases sont mélangées. Pour cela, vous pouvez utiliser l'algorithme suivant.

- Initialiser la grille dans la configuration gagnante.
- Tirer au sort un entier  $n$  entre 1 et  $p^2 - 1$  et échanger la  $(p^2 - 1)^\circ$  case avec  $n^\circ$  case. Recommencer pour la  $(p^2 - 2)^\circ$  case, en tirant un entier compris entre 1 et  $p^2 - 2$  et ainsi de suite jusqu'à avoir traité toutes les cases.

Pour le tirage au sort, un paquetage `paq_random` vous est fourni dans `/mnt/info/ada/touzet/`. La fonction `Random(Limite:Positive) return Positive` rend un entier choisi aléatoirement dans l'intervalle  $1..Limite$ .

**Question 3.** Écrire une procédure `Affichage` qui affiche la grille du taquin, avec des espaces pour la case libre.

**Question 4.** Écrire une procédure `Un_coup(T:in out Taquin;X,Y:Positive)` qui coulisse la case de coordonnées  $(X, Y)$  dans la case libre. Par exemple, jouer  $(3, 1)$  donne

5	6	3	→	5	6	3
1	4	7		1	4	7
8		2			8	2

**Question 5.** Écrire une fonction `Gagne(T:Taquin)` qui teste si le taquin est dans la configuration gagnante.

**Question 6.** Écrire la procédure `Jouer(T:in out Taquin)` pour faire une partie complète : initialisation, puis saisie des coups, jusqu'à gagner.