

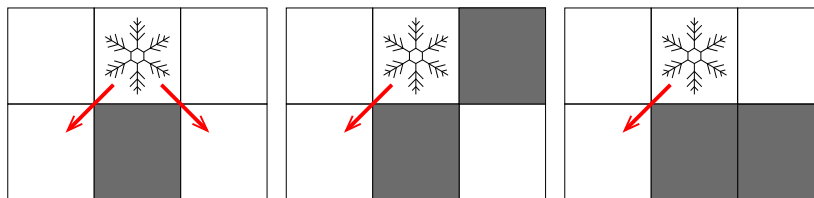
## TP n°1

### Tombe la neige (dans le terminal)

Le but de ce TP est de prendre en main la bibliothèque `ncurses` et de mettre en place l'interface du projet n°1.

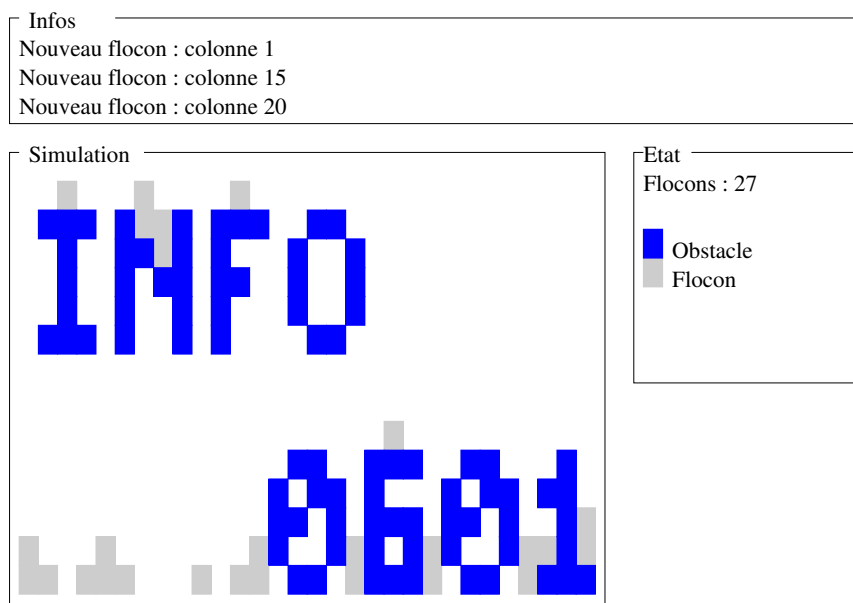
## 1 Tombe la neige : l'interface

L'objectif de l'application est de réaliser une simulation de flocons de neige. Les flocons sont lâchés en haut de l'écran et tombent jusqu'à rencontrer un obstacle ou un flocon de neige. Si le flocon peut tomber à droite ou à gauche de l'obstacle, il continue sa route à droite ou à gauche, en diagonale. Sinon, il s'arrête. La figure ci-dessous illustre les déplacements possibles en fonction des obstacles :



Nous souhaitons mettre en place une interface sous `ncurses` constituée de trois fenêtres (et trois sous-fenêtres) :

- La fenêtre d'informations : elle fait la largeur du terminal sur 3 lignes de haut (en plus du cadre) ; elle permet d'afficher des messages qui défileront lors de l'exécution de l'application.
- La fenêtre de simulation : elle fait 15 caractères de haut sur 30 de large.
- La fenêtre d'état : elle permet simplement d'afficher le nombre de flocons tombés et la légende.



1. Créez l'interface proposée en procédant à toutes les vérifications d'usage.



Utilisez la fonction `getch` pour mettre votre application en pause et observer votre interface.

2. Écrivez une fonction permettant d'afficher un message dans la fenêtre d'informations. Assurez-vous que le défilement est activé dans la fenêtre.
3. Écrivez une fonction pour mettre à jour le nombre de flocons.

## 2 Tombe la neige : la boucle principale

Nous souhaitons maintenant que l'application principale fasse tomber la neige. Pour cela, nous créons une zone de simulation (30x15) qui contient des obstacles générés aléatoirement.

1. Écrivez une fonction permettant de générer une zone de simulation aléatoire (des obstacles sont placés aléatoirement).
2. Écrivez une fonction qui affiche la zone de simulation.

La boucle principale de votre programme consiste à générer aléatoirement des flocons et à les faire tomber jusqu'à leur immobilisation. Si un obstacle est présent sur la première ligne de la colonne choisie, une nouvelle colonne doit être générée. Le programme doit tourner tant que l'utilisateur ne presse pas la touche F2.

3. Terminez l'application demandée.



- Rendez le `getch` non bloquant, ce qui permet de tester si l'utilisateur appuie sur une touche ou si le flocon en cours doit descendre ;
- Utilisez des variables vous permettant de savoir si un flocon est en cours de descente : si c'est le cas, le flocon tombe, sinon un nouveau flocon est généré.