

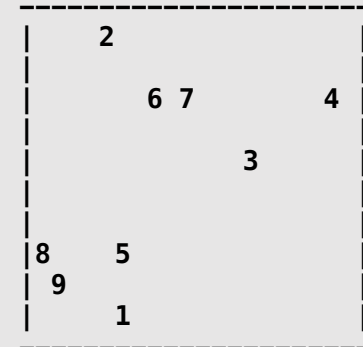
Thème

- Générateur aléatoire
- Classes et encapsulation
- Compilation séparée

But

Nous souhaitons simuler des robots autonomes dans un espace donné. L'utilisateur est invité à saisir les dimensions (X, Y) et le nombre de robots [1, 9].

```
ce programme ...  
largeur      [10..1000] : 20  
hauteur      [10..1000] : 10  
nbre object  [0..9] : 9
```



Les règles du jeu sont simples

- La partie commence en disposant les robots de manière aléatoire et sans recouvrement.
- Les numéros représentent les id des robots qui est naturellement unique.
- Les robots se déplacent de manière autonome d'une unité à la fois et dans une direction aléatoire (UP, DOWN, RIGHT, LEFT).
- Les robots restent dans l'espace choisi.
- Lorsqu'un robot prend la place d'un autre (même position), ce dernier est détruit et quitte la partie.
- La partie se termine lorsqu'un seul robot est en jeu.

Contraintes

- Déclarer et définir les classes nécessaires qui doivent respecter autant que possible le principe d'encapsulation.
- Utiliser au mieux tout ce qui a été vu à ce jour.
- Économiser la mémoire au maximum.
- La représentation doit correspondre à celle proposée.

Bibliographie / Support

Pour réaliser l'animation, la page du terminal est effacée avant l'affichage de la nouvelle représentation.

La méthode `this_thread::sleep_for` et la fonction `system("clear")` peuvent être utiles.