

# TPs 1

Master 1 SID

Benoist GASTON

[benoist.gaston@univ-rouen.fr](mailto:benoist.gaston@univ-rouen.fr)



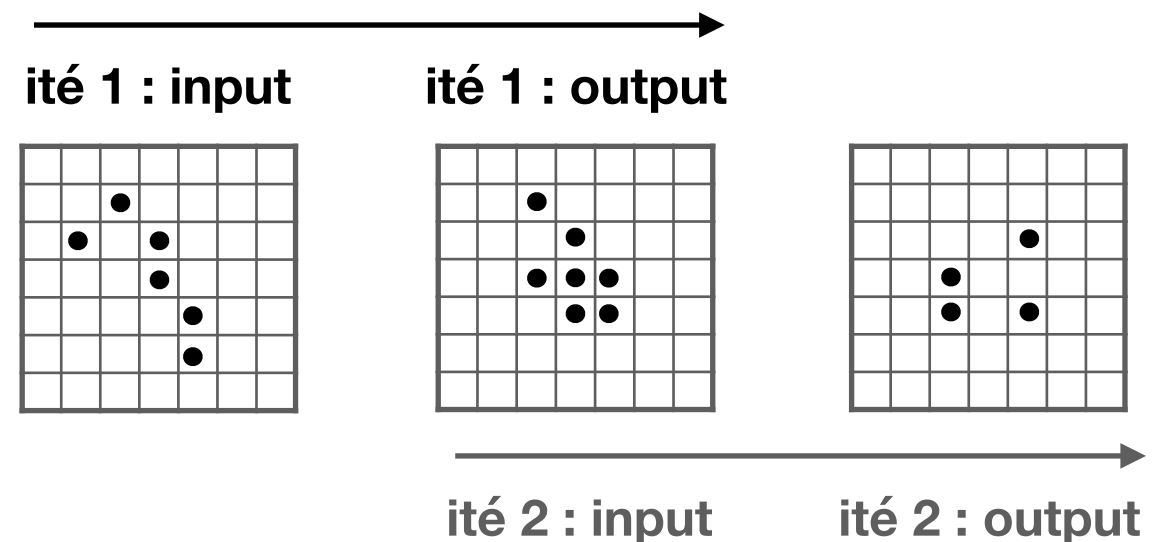
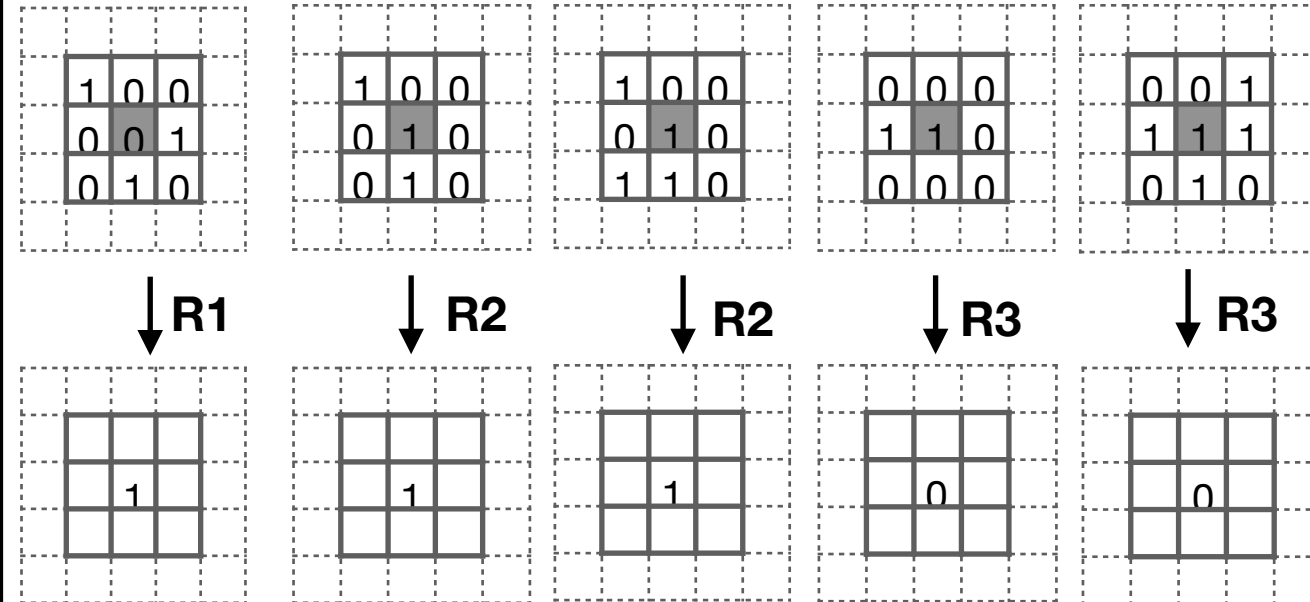
# Jeu de la Vie

## Présentation

- Les jeux de la vie, ou automates cellulaires, sont définis sur une grille de cellules. Les cellules sont dans un état donné (mort ou vivant). L'état des cellules évolue dans le temps en fonction de l'état des cellules voisines selon des règles simples.

## Règles de base

- Etat 0 ou 1, i.e. morte ou vivante
- R1 : une cellule morte possédant exactement 3 voisins (vivants) naît, i.e. Etat 0->1
- R2 : une cellule vivante possédant 2 ou 3 voisins (vivants) reste vivante, i.e. Etat 1->1
- R3 : une cellule vivante qui possède moins de 2 voisins (vivants) ou plus de 3 voisins (vivants) meurt par isolement ou surpeuplement i.e. Etat 1->0



# Jeu de la Vie

On vous propose dans ces TPs de réaliser une implémentation efficace du Jeu de la Vie

Pour cela vous utiliserez comme matière première le fichier `masterjdv.py` (disponible sur <https://github.com/benoistgaston/m1sid-2020.git> TPs/JdV) qui contient trois fonctions de base

## 1. La fonction `init_grid(n)`

- prend en entrée une dimension `n`
- retourne une grille 2D de dimension `n x n` rempli aléatoirement de 0 et de 1

## 2. La fonction `get_nbneigh(grid, coord)`

- prend en entrée une grille `grid` et un couple coord de coordonnée `i, j`
- retourne la liste des valeurs des voisins de `i, j` dans la grille

## 3. Une fonction `evolution1(grid)`

- prend en entrée une grille `grid`
- retourne une grille 2D qui est l'évolution de la grille `grid` en application des 3 règles énoncées

## Questions

1. Appliquer la fonction `evolution1` et estimer les temps de calcul pour des grilles de tailles 10x10 100x100 1000x1000
2. Utiliser `cProfile` afin d'identifier les fonctions les plus coûteuses (sur des grilles 2000x2000)
3. Créer une fonction `evolution1_corr` qui est une intégration du corps de `get_nbneigh` dans `evolution1`
4. Utiliser de nouveau `cProfile` afin d'estimer un potentiel gain
5. À partir de `evolution1_corr`, écrire une fonction `evolution1_ndarray` qui modifie la structure de données de `res_grid` afin d'utiliser un `ndarray` 2D
6. Quels impacts sur les performances du code ?