

Jeux de la vie

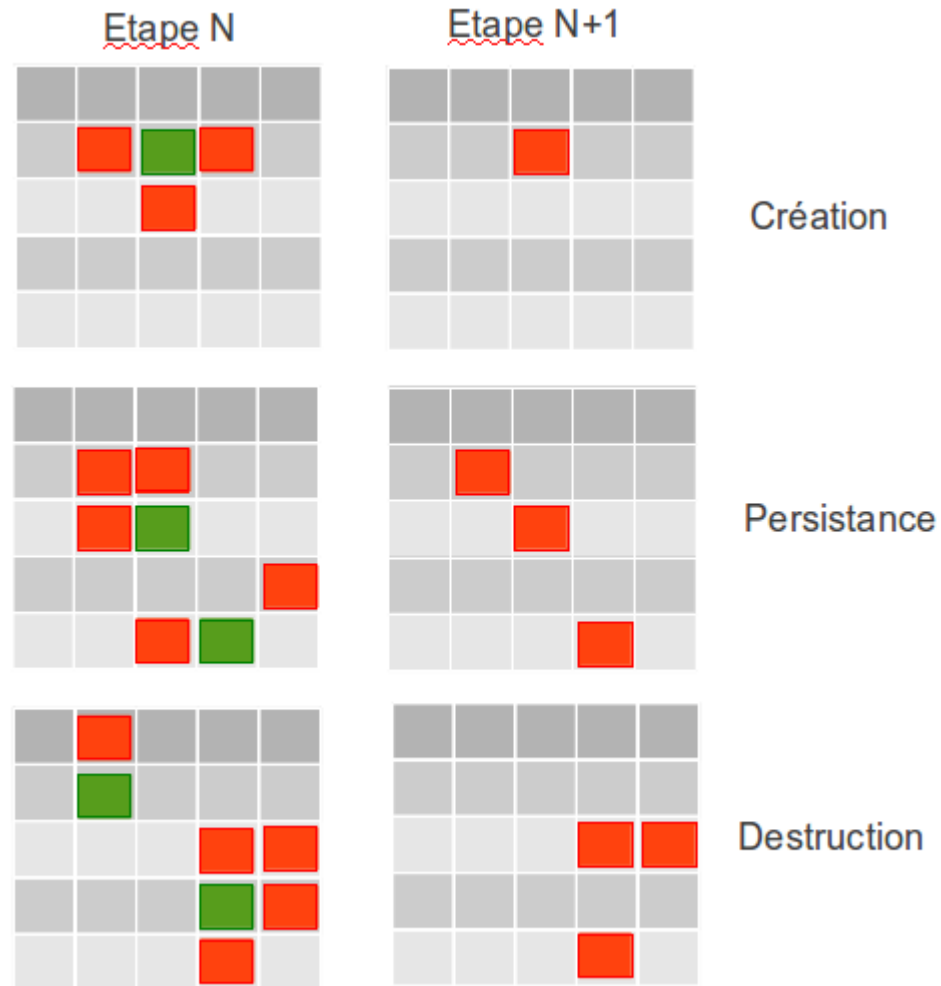
# Table des matières

- Présentation du jeux de la vie
- Les algorithmes de calcule
  - Linéaire
  - Algorithme « threadé »
    - Barrière
    - Découpage de la zone de jeux.

# Jeux de la vie

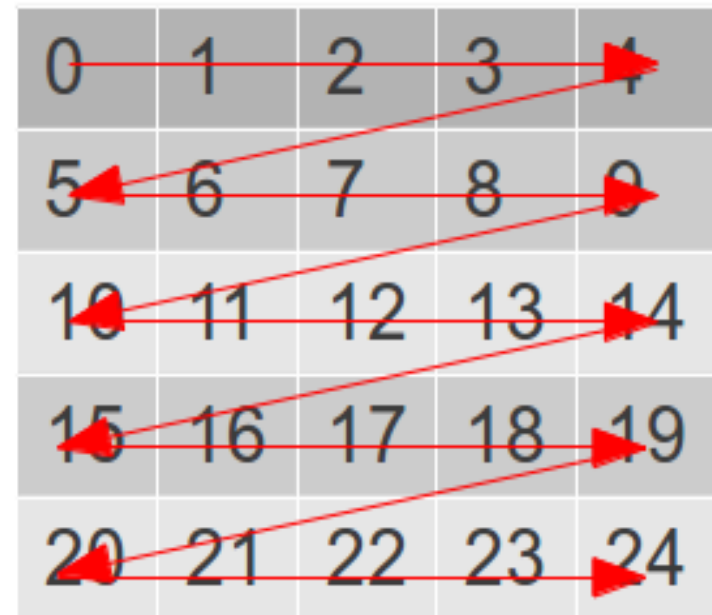
- Grille de taille quelconque
- Calcul du nombre de voisin
  - 3  $\rightarrow$  case prend vie
  - 2 et vivante au tour d'avant  $\rightarrow$  reste vivante
  - $> 3 \rightarrow$  meurt par surpopulation
  - $\leq 2$  meurt par sous population

# Jeux de la vie



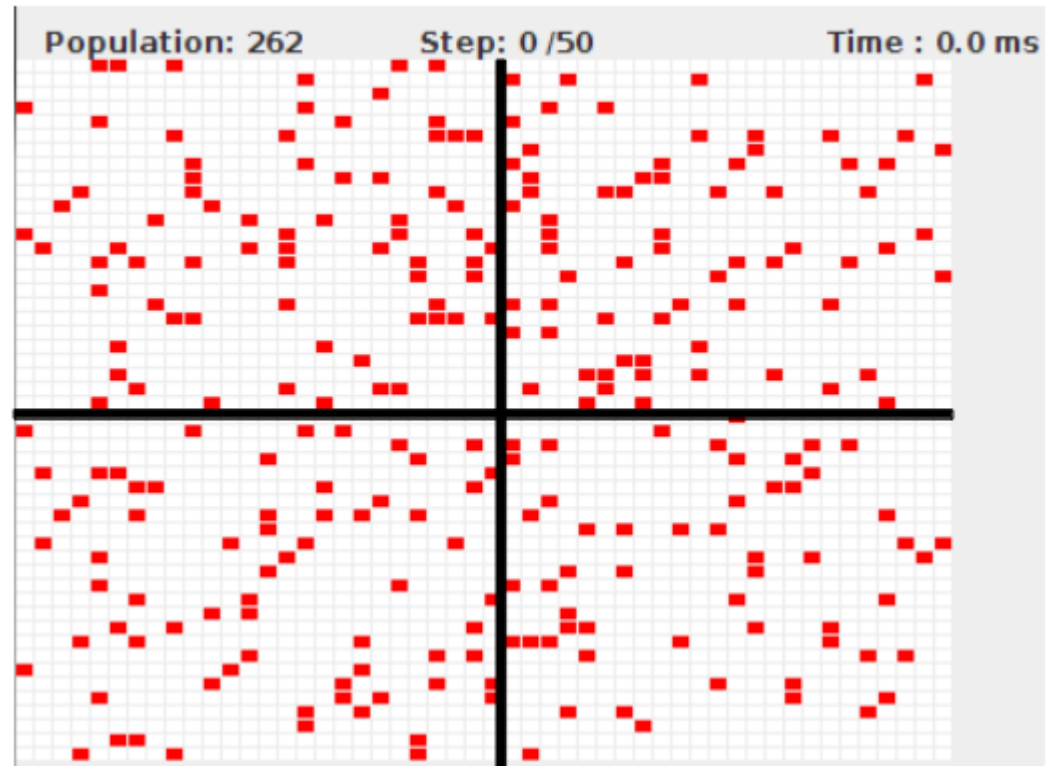
# Algorithme linéaire

- Algorithme simple
- Efficacité pour un grand nombre de case

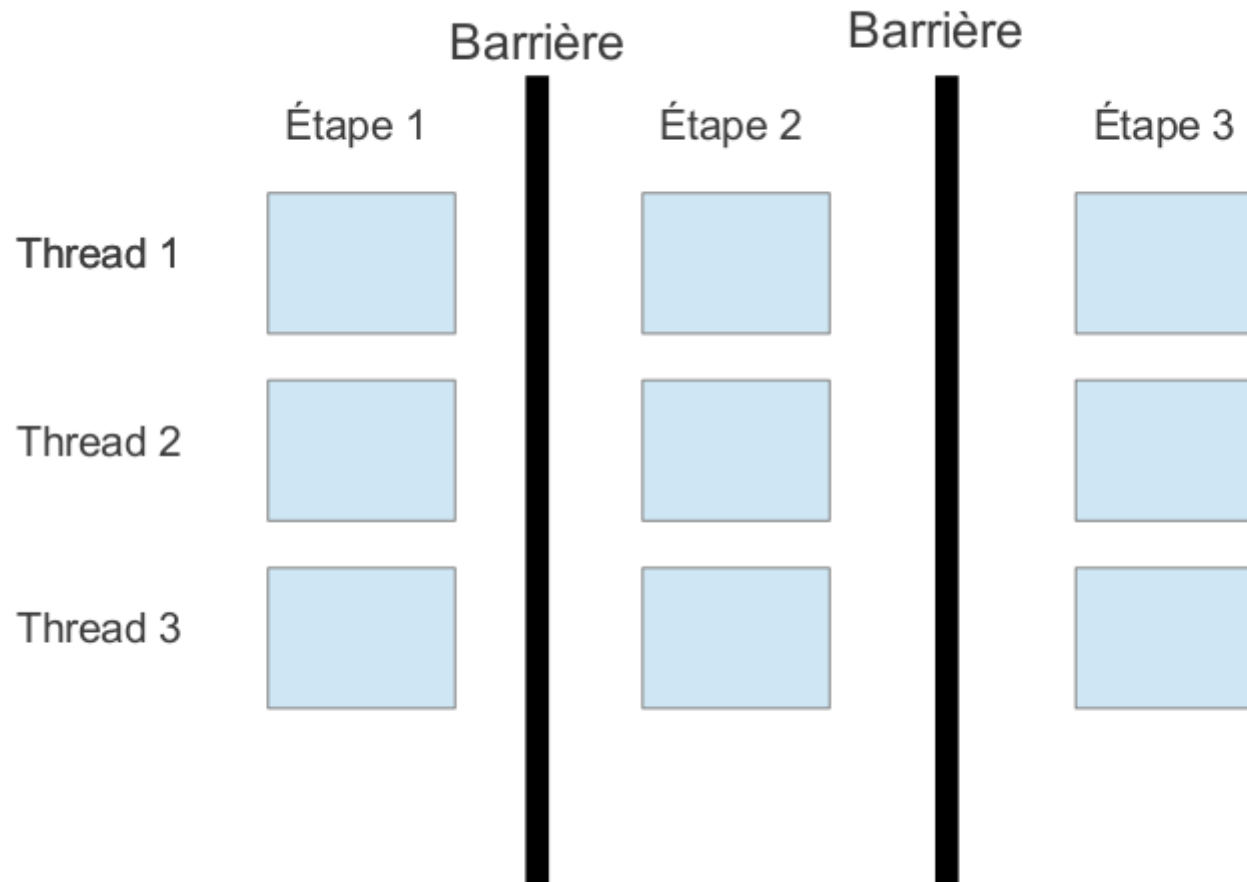


# Algorithme « threadé »

- Découpage de la zone de jeux en 4.
- Création de 4 threads
- Synchronisé avec des barrières



# Les barrières



# Découpage en N

- Extension de l'algorithme précédent.
- Choix du nombre de case par thread.



# Un thread par case

- Temps de création des thread très lent.

# Nombre maximum de thread

Test sur 40 000 cases

33 050 theads au maximum

Exception in thread "AWT-EventQueue-0" java.lang.OutOfMemoryError: unable to create new native thread

at java.lang.Thread.start0([Native Method](#))

at java.lang.Thread.start([Thread.java:640](#))

at java.awt.EventQueue.initDispatchThread([EventQueue.java:878](#))

at java.awt.EventDispatchThread.run([EventDispatchThread.java:153](#))