

FEUX DE FORÊTS ET AUTOMATES CELLULAIRES

Quentin Huan

I Présentation du sujet

Dans ce module, nous allons réaliser une simulation de feu de forêt à l'aide d'un modèle simple :

- la forêt est représentée par une grille $N \times N$, $N \in \mathbb{N}$
- chaque case de la grille peut prendre 3 valeurs :
 - 0 si la case est vide
 - 1 si la case contient un arbre
 - 2 si la case contient un arbre en feu
- une étape de simulation consistera à mettre à jour la grille en suivant 2 règles :
 - une case en feu au tour $[t]$ est remplacée par une case vide au tour $[t + 1]$
 - un arbre adjacent à une case en feu au tour $[t]$ à une probabilité p de s'enflammer au tour $[t + 1]$

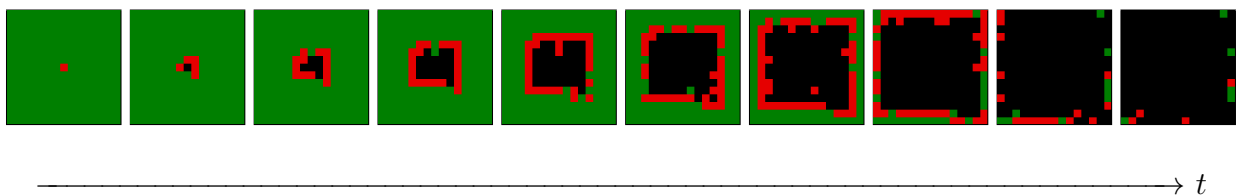


FIGURE 1 – Exemple de simulation, $p = 0.8$

II Organisation

Volume : 9*3h

Evaluation : projet à rendre à la fin de la dernière séance (envoyer une archive par binôme avec noms/prénoms/groupe à quentin.huan@univ-littoral.fr)

La notation se fera sur :

- nombre de fonctionnalités implémentées
- qualité du code (organisation, commentaires, lisibilité...)
- + devoir sur table (à confirmer)

Le travail pourra se faire en binôme.

La séance 1 se fera sur papier (comprehension du problème, mise en place du squelette du programme, choix des structures de données...).

III Objectifs principaux

A Développement de la simulation

Dans un premier temps :

1. **réaliser la simulation** en *Python*, en affichant chaque étape dans la console.
2. On pourra ensuite se servir du module *Pil* pour **enregistrer des images** sur le disque (du même genre que Figure 1).

i Documentation

Pour enregistrer des images :

- [liens vers la documentation de PIL](#)
- [fonction Image.fromArray](#)
- [fonction Image.save](#)

B Mise en évidence d'un effet de seuil

On remarquera que la simulation peut se comporter de deux manières suivant la valeur de la probabilité p :

- si p est faible, l'incendie brûle quelques arbres, mais il finit par s'éteindre
- au dessus d'un certain seuil, l'incendie devient incontrôlable et brûle toute la forêt

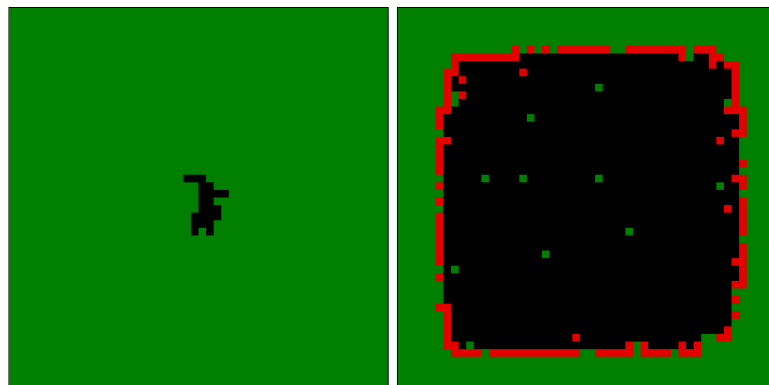


FIGURE 2 – Illustration des deux comportements : à gauche $p = 0.2$, à droite $p = 0.8$

On aimerait :

- savoir comment évolue la proportion d'arbres brûlés en fonction de p
- mesurer le seuil p^* au dessus duquel la forêt entière brûle

Si le besoins s'en fait sentir, on utilisera le module **Matplotlib** pour réaliser des graphiques.

i Documentation

[\[liens vers la documentation de Matplotlib\]](#)

IV Pour aller plus loin...

Ces parties sont indépendantes et ne suivent pas d'ordre particulier.

A Améliorations de la simulation : performance

On remarquera que notre programme devient lent lorsque N est grand :

- quelle est la complexité de votre programme ?
- proposer une solution pour l'améliorer

B Améliorations de la simulation : effets supplémentaires

Le modèle utilisé était simpliste, mais on peut ajouter quelques comportements pour le rendre plus réaliste. Quelques idées...

- varier la distribution des arbres sur la grille
- modéliser le vent (la propagation est favorisée dans une direction)
- rendre certaines zones de la forêt plus difficile à brûler (zone plus humide ?)
- ...

C Améliorations de l'affichage : affichage en temps réel

On pourrait réaliser un affichage en temps réel de la simulation et ainsi permettre à l'utilisateur d'interagir avec (planter des arbres, mettre le feu, mettre la simulation en pause, ...).

(se renseigner sur les modules *Pygame* ou *Pyglet*).

D Les automates cellulaires

Le programme que nous venons d'implémenter appartient à une classe de programme plus vaste que l'on appelle les **automates cellulaires**. Il est possible de changer la règle de mise à jour pour obtenir toute une variété de comportements. Quelques règles emblématiques (ou juste amusantes) que l'on peut implémenter :

- le Jeu de la Vie de *John Conway* (voir [\[ici\]](#)) → facile
- génération procédurale de cavernes (voir [\[ici\]](#)) → facile
- règle "pierre-feuille-ciseaux" → facile
- les automates cellulaires élémentaires (voir [\[ici\]](#))
- simulation de fluides pour les jeux-vidéo (regarder [\[par-là\]](#))

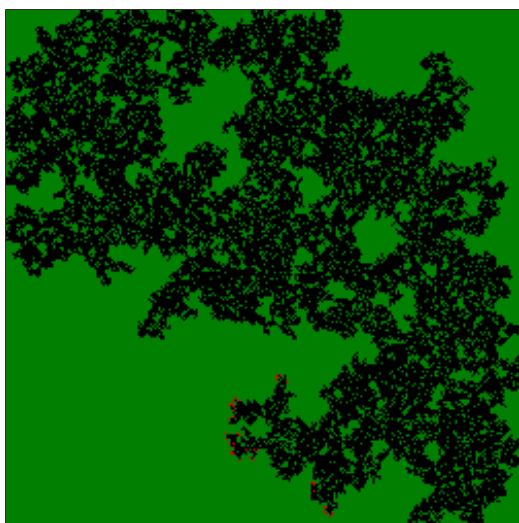
- génération procédurale de niveaux dans les jeux-vidéo (voir [ici](#)) \rightarrow *facile*
- ...

Remarque : (les items marqués " \rightarrow *facile* " ne nécessitent qu'une petite modification du programme pour fonctionner)

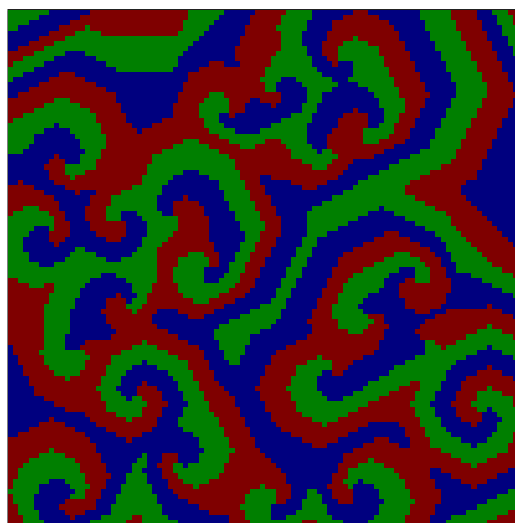
E Percolation

Notre problème initial peut se ramener à un problème de **percolation** : voir [\[ici\]](#) un article du site *images des Mathématiques*

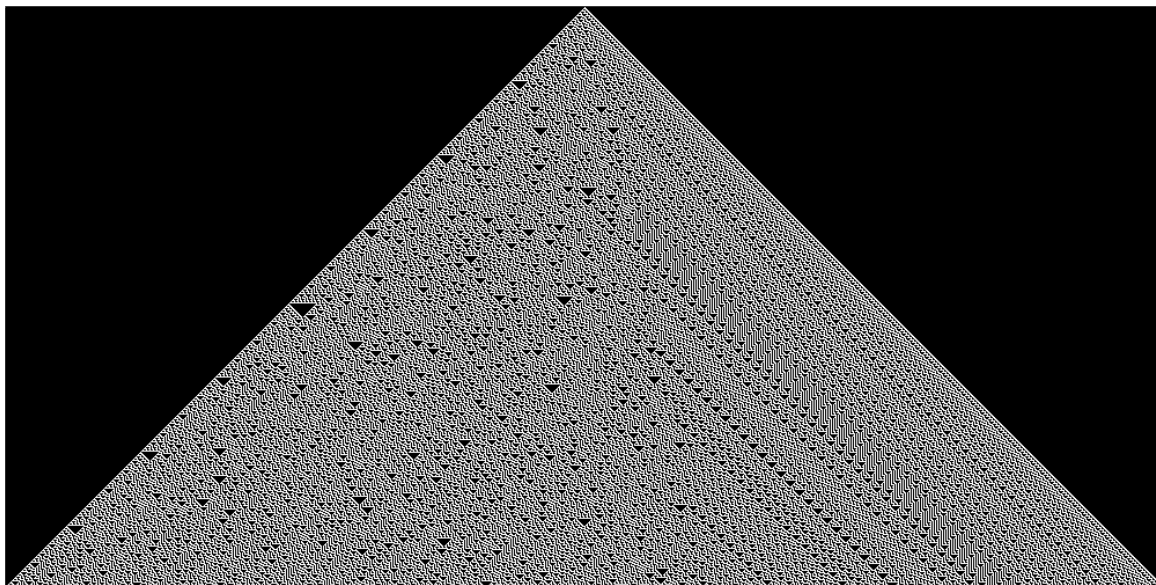
F Illustrations



(a) Simulation d'incendie



(b) Règle "pierre-feuille-ciseaux"



(a) Automate élémentaire, règle 86