



Simulation de propagation d'un feu de forêt

Modélisation au moyen d'automates cellulaires

Hamdi Gazzeh

Julian Lock

Nathan Oiknine

Delphine Salzmann

Utilité d'une telle modélisation

- Recrudescence des incendies, par exemple en Californie, Australie, France, etc...
- Modélisation pour étudier la propagation du feu dans une forêt au contact d'habitation
- Etude de l'impact du vent, de la pluie, des pompiers sur cette propagation
- Modélisation de propagation à l'aide d'automates cellulaires



Découpage et répartition du travail

Sprint 0 : Réflexion

- réflexion autour de l'objectif du projet et choix de sa modélisation
- établissement des règles (nature des cases, règles de propagation, ...)
- découpage en fonctionnalités
- début rédaction du README

Sprint 1 : Mise en place du support de propagation

- Fonctionnalité 1 : générer la grille de paysage sous forme matricielle en associant à chaque case une liste [etat, nature] --> fonction grille()
- Fonctionnalité 2 : Afficher le paysage --> fonction paysage()
- Fonctionnalité 3 : initialiser un départ de feu --> fonction depart()

Sprint 2 : Effet du vent

- Fonctionnalité 4 : définir un facteur de vent selon sa force et sa direction --> fonctions fact_vent_x_plus, fact_vent_x_moins, fact_vent_y_plus, fact_vent_y_moins

Sprint 3 : Simulation de propagation de feu

- Fonctionnalité 5 : Configurer les probabilités de combustion et d'extinction de feu liées à une case --> dictionnaires probabilite_de_combustion et probabilite_feu_eteint
- Fonctionnalité 6 : Déterminer les voisins en feu d'une case --> fonction voisins()
- Fonctionnalité 7 : Appliquer les règles de propagation pour une case --> fonction propagation()
- Fonctionnalité 8 : Appliquer les règles de propagation à toutes les cases du paysage --> fonction generation_suivante()
- Fonctionnalité 9 : Simuler la propagation jusqu'à satabilisation de la situation --> fonction simulation()

Découpage et répartition du travail

Sprint 4 : Interface graphique

- Fonctionnalité 10 : utiliser Tkinter pour relever un départ de feu lorsqu'on clique sur la case
- Fonctionnalité 11 : compléter l'interface Tkinter avec des boutons et une légende pour afficher les générations successives

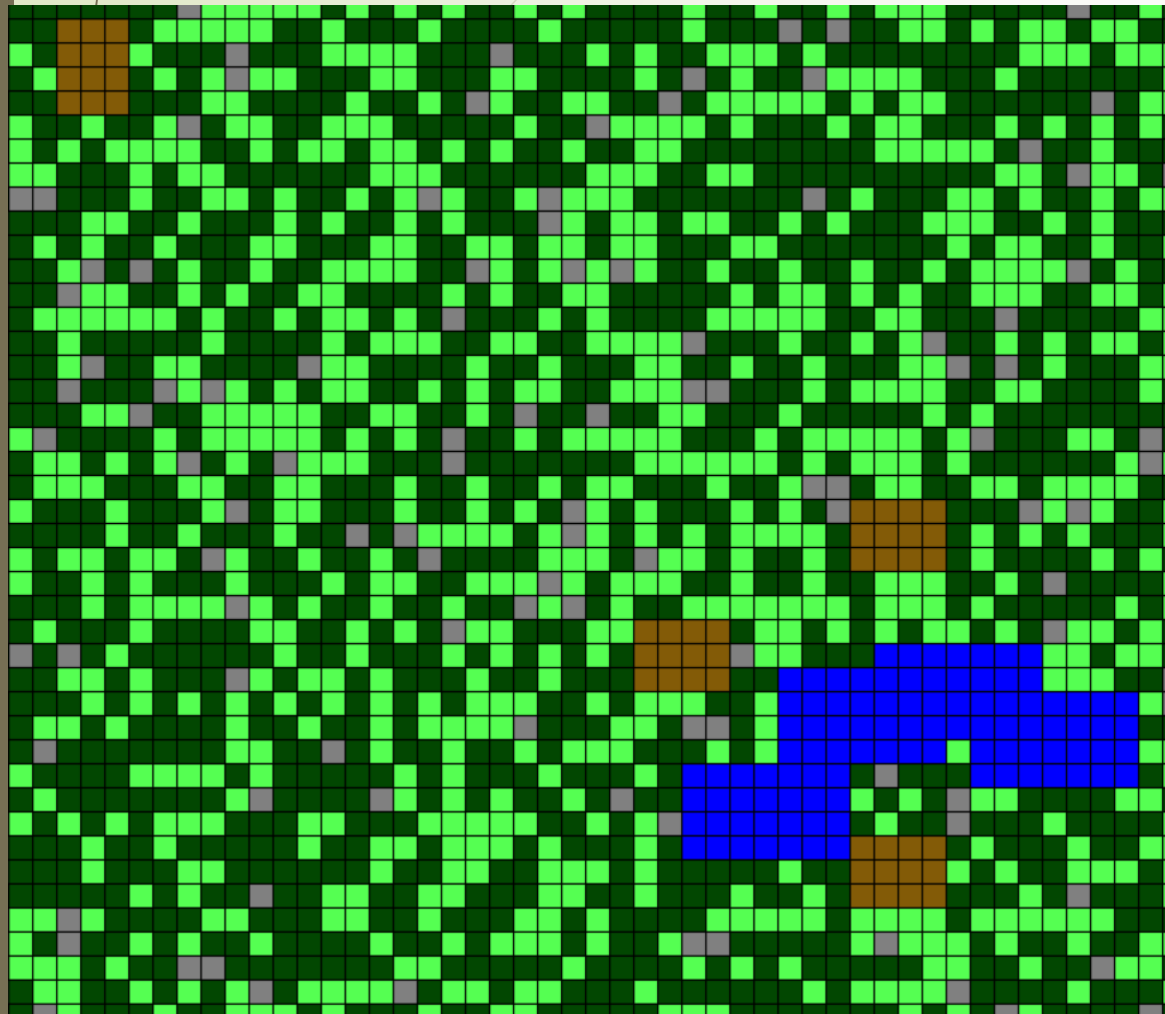
Sprint 5 : Effet de la chaleur

- Fonctionnalité 12 : une case est désormais caractérisée par une liste [état, nature, chaleur] où chaleur est un entier positif --> grille_c()
- Fonctionnalité 13 : une case chaude est de couleur orange --> paysage_c()
- Fonctionnalité 14 : les règles de propagation sont affinées en conséquent: à proximité du feu, le coefficient de chaleur est augmenté de 1 et sa probabilité de brûler à la génération suivante augmentée --> fonction propagation_c()
- Fonctionnalité 15 : écrire la fonction génération suivante avec ces nouvelles règles --> fonction génération_suivante_c()

Sprint 6 : Effet de la pluie

- Fonctionnalité 16 : Générer une grille de pluie sous forme matricielle en associant à chaque case un float entre 0 et 1 représentant l'intensité de la pluie --> fonction début_pluie()
- Fonctionnalité 17 : Faire évoluer une grille de pluie sous l'effet du vent --> fonctions évolution_pluie_vent () et évolution_pluie()
- Fonctionnalité 18 : Modifier la grille de la forêt en fonction de la pluie selon les règles suivantes: si une case est en feu, ce feu s'éteint avec comme probabilité l'intensité de la pluie sur cette case et sinon elle devient froide --> fonction effet_pluie()

Explication du jeu



Arbre
Herbe sèche
Eau
Habitation
Rocher

➤ **Propagation probabiliste**

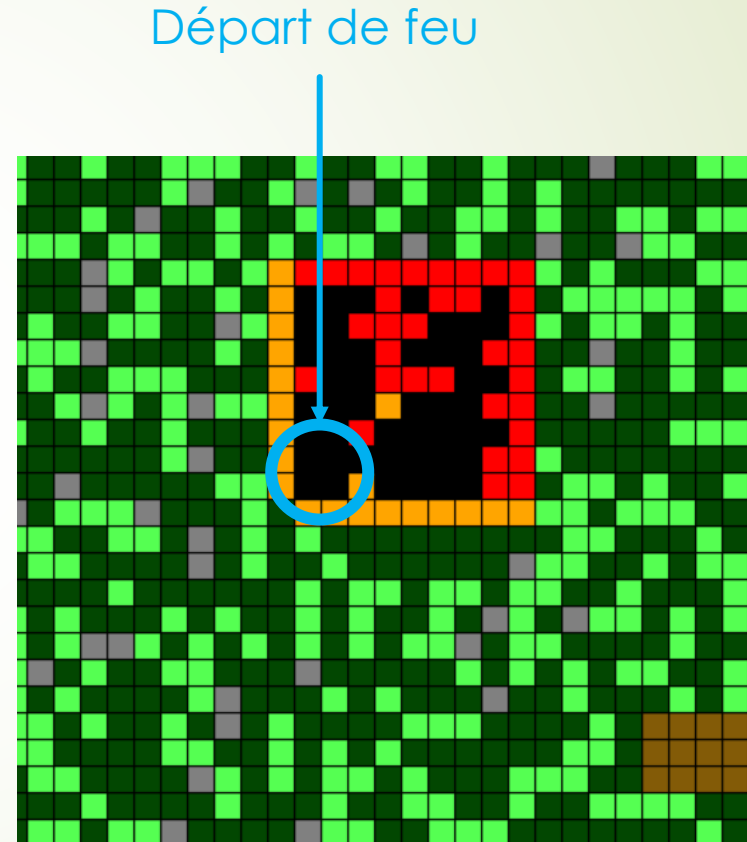
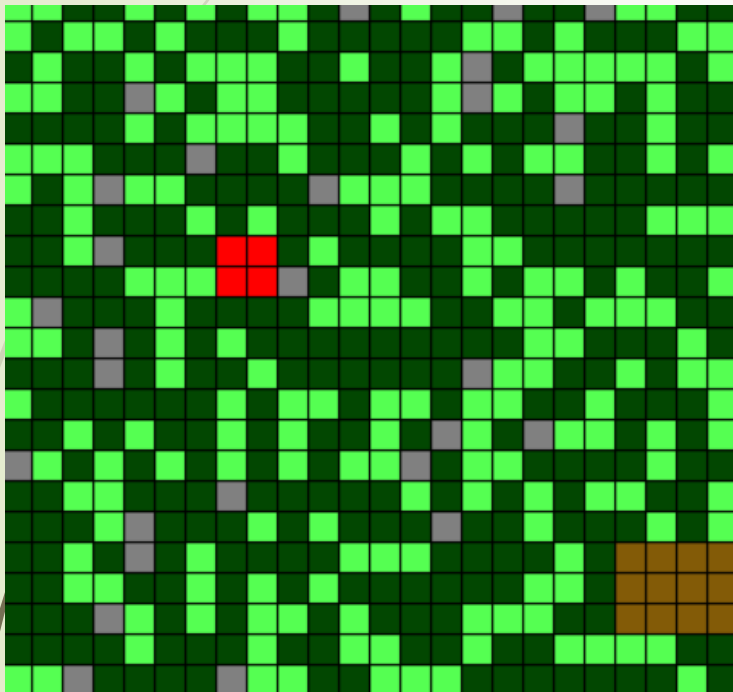
Probabilité de combustion :

Herbe sèche > arbre > maison > rocher > eau

Probabilité d'extinction du feu :

Rocher > herbe sèche > arbre > maison

Effet du vent



Effet d'un vent vers le Nord-Est
après quelques générations

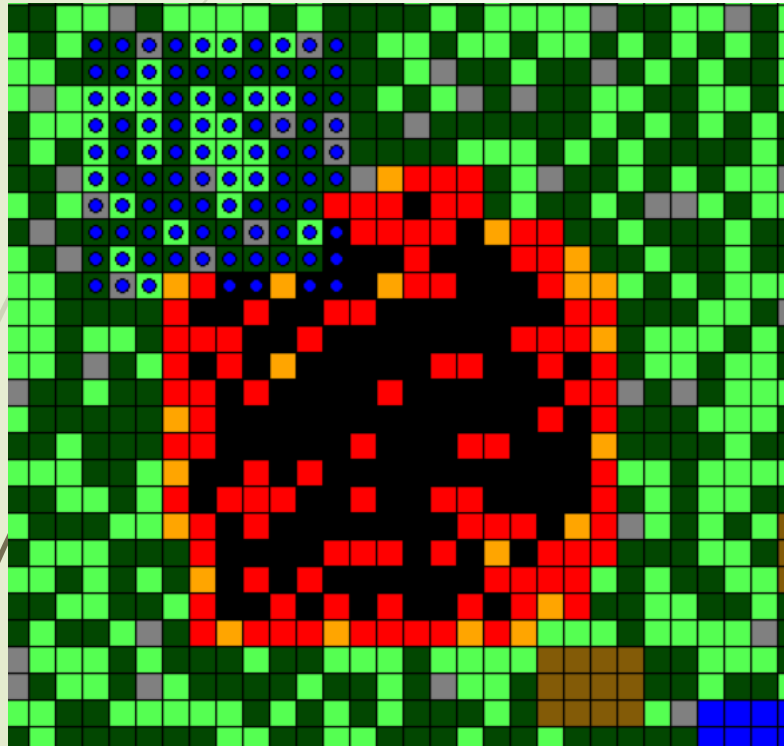
Prise en compte d'une propagation de chaleur



➤ Une case : [état, nature, chaleur]

- État {0,1}
- Nature {0,1,2,3,4,5}
- Chaleur = entier positif

Effet de la pluie



- Ajout d'un nuage de pluie qui se déplace dans le sens du vent

Couverture

```
$ py -m pytest --cov=incendies-en-foret --cov-report html test_*.py
===== test session starts =====
platform win32 -- Python 3.9.13, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\julia\OneDrive\Documents\CW-Semaine2\incendies-en-foret
plugins: cov-4.0.0
collected 14 items

test_depart_feu.py . [ 7%]
test_generation.py . [ 14%]
test_paysage.py .. [ 28%]
test_pluie.py .... [ 57%]
test_probabilites.py .... [ 85%]
test_propagation.py .. [100%]
C:\Users\julia\AppData\Local\Programs\Python\Python39\lib\site-packages\coverage\inorout.py:519: CoverageWarning: Module incendies-en-foret was never imported. (module-not-imported)
  self.warn(f"Module {pkg} was never imported.", slug="module-not-imported")
C:\Users\julia\AppData\Local\Programs\Python\Python39\lib\site-packages\coverage\control.py:801: CoverageWarning: No data was collected. (no-data-collected)
  self._warn("No data was collected.", slug="no-data-collected")
WARNING: Failed to generate report: No data to report.

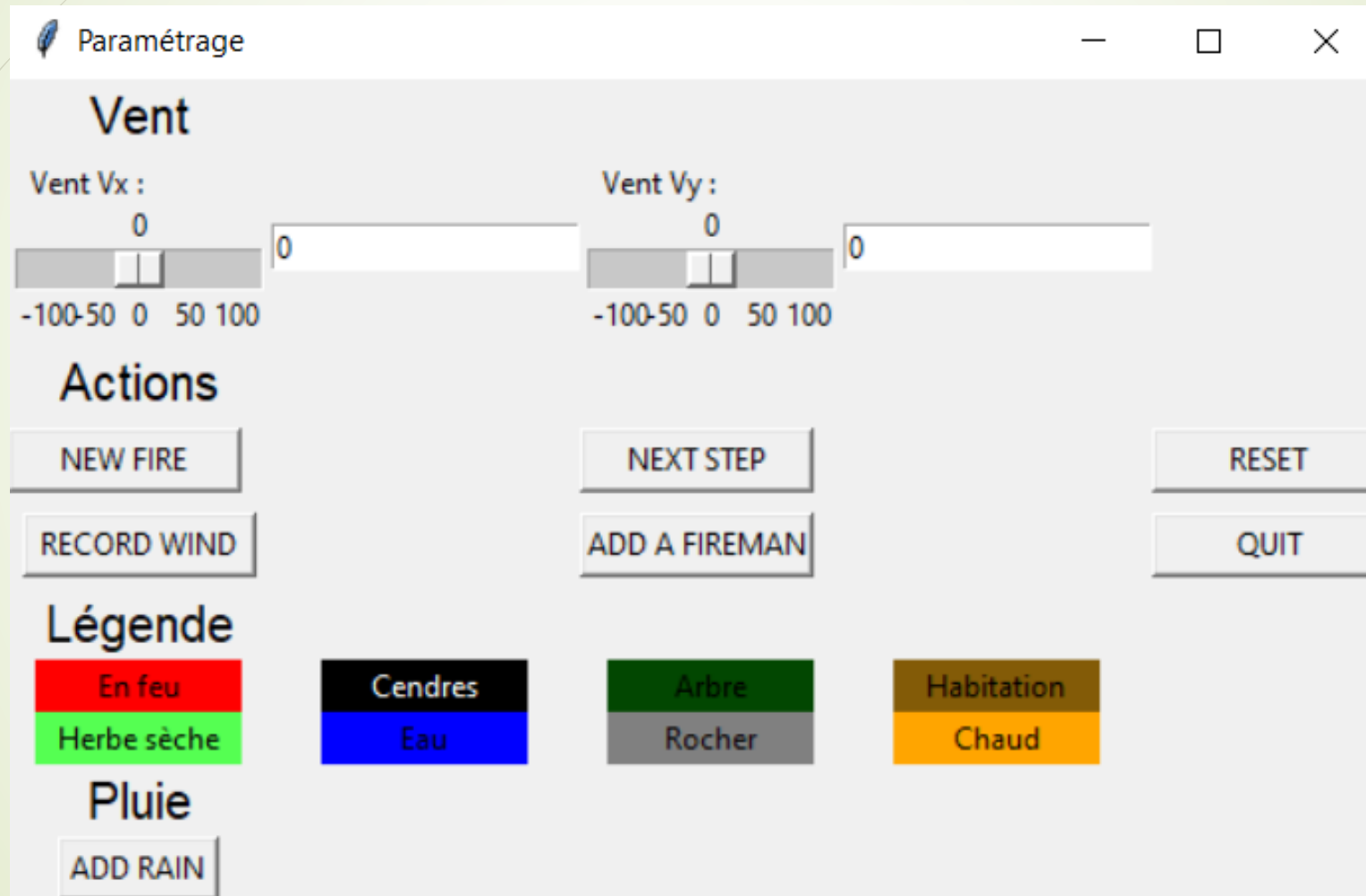
C:\Users\berna\AppData\Local\Programs\Python\Python310\lib\site-packages\pytest_cov\plugin.py:311: CovReportWarning: Failed to generate report: No data to report.

  warnings.warn(CovReportWarning(message))

----- coverage: platform win32, python 3.10.6-final-0 -----

===== 14 passed in 8.03s =====
```

Une interface Tkinter





Bilan

- Cohérence des probabilités et facteurs, mais pour une étude plus poussée, il faudrait faire des relevés sur le terrain
- Modélisation de la propagation du feu et de la chaleur, de l'effet du vent, de l'effet de la pluie réaliste