

## Projet en python<sup>™</sup> : simulateur de propagation d'incendie

### 1. Description

Le but du projet est la simulation de propagation d'un incendie, basée sur un automate cellulaire à deux dimensions (e.g. le *Jeu de la Vie*, ou *Game of Life*, de John Conway). Celui-ci évolue en fonction de différents paramètres comme la nature du terrain ou les conditions climatiques.

Le projet se décompose en plusieurs étapes : construction d'un terrain, affichage, simulation simple, et enfin, simulation avec probabilités et conditions climatiques.

### 2. Terrain

La simulation se déroule sur un environnement rectangulaire vu du dessus. Celui-ci se subdivise en parcelles carrées, chacune contenant un terrain particulier. On en distingue initialement trois types : eau, plaine, forêt. Une parcelle de tout type de terrain peut se transformer en parcelle calcinée une fois qu'elle a brûlé durant un certain temps.

Initialement, l'environnement est généré aléatoirement selon l'information suivante : chaque type de terrain n'a pas la même probabilité  $P$  d'apparaître :  $P(\text{forêt}) > P(\text{plaine}) > P(\text{eau})$ . Des maisons sont également réparties aléatoirement dans l'environnement et se substituent à l'information de la nature du terrain. Les densités des parcelles de chaque type sont données en paramètres. Au début de la simulation, un feu démarre sur une parcelle de forêt ou plaine.

Dans la réalité, les forêts (resp. plaines et eau) forment des zones homogènes (contiguës). Par exemple, les zones d'eau forment des rivières ou des lacs.

### 3. Propagation

Une plaine brûle plus vite qu'une forêt, qui brûle plus vite qu'une maison. Ainsi une maison brûle en 8 niveaux d'intensité de feu ( $\text{intensiteMax} = 8$ ), une forêt brûle en 4 niveaux d'intensité de feu ( $\text{intensiteMax} = 4$ ), et une plaine brûle en 2 niveaux d'intensité de feu ( $\text{intensiteMax} = 2$ ).

Une parcelle qui prend feu, brûle à chaque unité de temps jusqu'à son intensité maximale, puis le feu s'estompe petit à petit en voyant son intensité diminuer jusqu'à 1. Après avoir brûlé, la parcelle devient calcinée à la prochaine unité de temps. À chaque étape, les prochains états des parcelles sont calculés en fonction de règles de calcul, données par la suite.

La propagation s'arrête lorsqu'aucune parcelle n'a changé d'état après une unité de temps. Les deux simulations suivantes devront être considérées.

#### 3.1. Simulation simple

Les règles de calcul de la simulation simple sont les suivantes :

1. Une parcelle de type eau ne brûle jamais.
2. Une parcelle de type forêt, plaine ou maison ayant une de ses 8 voisines en feu, prend feu d'intensité 1.
3. Un feu qui brûle d'une intensité  $i < \text{intensiteMax}$  une parcelle, incrémente son intensité de 1 à la prochaine unité de temps.
4. Un feu brûlant d'une intensité  $i = \text{intensiteMax}$  une parcelle, s'estompe et décrémente son intensité de 1.
5. Un feu d'intensité  $i > 1$  qui s'estompe, voit son intensité passer à  $i - 1$ .
6. Un feu d'intensité 1 qui s'estompe sur une parcelle rend celle-ci calcinée.

### 3.2. Simulation avec probabilité

Dans cette deuxième simulation, nous allons prendre en compte la probabilité qu'une parcelle en feu influence la propagation de l'incendie sur les parcelles voisines. Dans cette situation, une parcelle de type forêt (ou plaine, ou maison) qui a une parcelle voisine en feu, peut rester intact. La probabilité varie en fonction de la distance, de l'intensité du feu et du type de terrain de la parcelle voisine.

#### 3.2.1 Règles de calcul

Les règles de calcul pour la deuxième simulation sont les suivantes :

1. Une parcelle de type eau ne brûle jamais.
2. Une parcelle de type forêt, plaine ou maison a  $P\%$  de chance de prendre feu d'intensité 1 pour chacune de ses 8 voisines, avec  $P = N \times M \times (0,75)^{(intensiteMax-k)}$  et où :
  - $N$  représente le facteur type de terrain.  $N = 0,5$  si la parcelle voisine est de type plaine ou maison en feu,  $N = 1$  si la parcelle voisine est de type forêt en feu, et  $N = 0$  sinon.
  - $M$  représente le facteur distance.  $M = 25$  si la parcelle voisine se trouve sur l'une des 4 diagonales de la parcelle  $(i, j)$ , et  $M = 75$  si la parcelle voisine se trouve à gauche, ou à droite, ou en haut, ou en bas de la parcelle  $(i, j)$ .
  - $k$  représente le facteur intensité de la parcelle voisine ( $k = 0$  si la parcelle voisine n'est pas en feu).

Exemple. Lorsque la parcelle  $(i, j)$  est de type forêt et qu'elle a deux de ses voisines en feu tel que la parcelle  $(i, j + 1)$  de type maison qui brûle avec intensité de 4 et la parcelle  $(i + 1, j + 1)$  de type plaine qui brûle avec une intensité de 2. La parcelle  $(i, j)$  a  $P_1 = 0,5 \times 75 \times 0,75^{(8-4)} = 11,86\%$  de brûler due à la parcelle  $(i, j + 1)$ . La parcelle  $(i, j)$  a  $P_2 = 0,5 \times 25 \times 0,75^{(4-2)} = 7,03\%$  de brûler due à la parcelle  $(i + 1, j + 1)$ .

	$j - 1$	$j$	$j + 1$
$i - 1$	$F$	$F$	$F$
$i$	$E$	$F$	$M^4$
$i + 1$	$E$	$F$	$P^2$

$\Rightarrow$

	$j - 1$	$j$	$j + 1$
$i - 1$	0	0	0
$i$	0	$(i, j)$	11,86
$i + 1$	0	0	7,03

FIGURE 1 – Exemple de calcul des probabilités :  $F$  pour forêt,  $E$  pour eau,  $P$  pour plaine,  $M^4$  pour maison brûlant d'intensité 4, et  $P^2$  pour plaine brûlant d'intensité 2.

3. Un feu qui brûle d'une intensité  $i < intensiteMax$  une parcelle, incrémente son intensité de 1 à la prochaine unité de temps.
4. Un feu brûlant d'une intensité  $i = intensiteMax$  une parcelle, s'estompe et décrémente son intensité de 1.
5. Un feu d'intensité  $i > 1$  qui s'estompe, voit son intensité passer à  $i - 1$ .
6. Un feu d'intensité 1 qui s'estompe sur une parcelle rend celle-ci calcinée.

#### 3.2.2 Conditions climatiques

Nous allons maintenant prendre en compte l'influence du vent, de la pluie et des éclairs sur la propagation de l'incendie.

**Vent.** Le vent souffle dans quatre directions possibles : gauche vers droite, droite vers gauche, haut vers bas et bas vers haut. Lorsque que le vent souffle de la droite vers la gauche, il augmente la probabilité  $P$  calculée dans la section 3.2.1 de  $P \times v$  où  $v$  est un paramètre qui prend ses valeurs dans le tableau de la FIG. 2(a) (le calcul de l'influence du vent pour les autres directions se fait de façon similaire).

Une parcelle  $(i, j)$  qui n'est pas en feu, a donc plus de chance de prendre feu par rapport à la parcelle  $(i, j - 1)$  qui est en feu et 0% de prendre feu par rapport à la parcelle  $(i, j + 1)$  parce que le vent souffle de la gauche vers droite. La parcelle  $(i, j + 1)$  qui est en feu n'influence donc pas sa parcelle voisine de gauche. Le calcul de l'influence du vent pour les autres directions se fait de façon similaire.

En l'absence de vent, toutes les parcelles du tableau valent 0. La présence du vent est gérée aléatoirement, et le vent souffle pendant un certain nombre de tours.

	$j - 1$	$j$	$j + 1$
$i - 1$	0,25	0	-1
$i$	0,25	0	-1
$i + 1$	0,25	0	-1

(a) Paramètre  $v$  (vent soufflant de droite à gauche).

	$j - 1$	$j$	$j + 1$
$i - 1$	0,25	0,25	0,25
$i$	0,25	0,25	0,25
$i + 1$	0,25	0,25	0,25

(b) Paramètre  $p$  (pluie).

FIGURE 2 – Influence des conditions climatiques sur la probabilité.

**Pluie.** La pluie quant à elle diminue la probabilité  $P$  de  $P \times p$  où le paramètre  $p$  prend ses valeurs dans le tableau de la FIG. 2(b). En l'absence de pluie, toutes les parcelles du tableau valent 0. La présence de pluie est gérée aléatoirement, et dure un certain nombre de tours.

La nouvelle probabilité en prenant compte le vent et la pluie est  $P_1 = P + P \times v - P \times p$ .

**Foudre.** Tous les 10 tours, la foudre peut tomber sur une parcelle. La foudre touche aléatoirement une parcelle ne contenant pas d'eau et ne brûlant pas, et entraîne sur celle-ci un nouveau foyer d'incendie.

Les conditions climatiques doivent être visibles à tout moment en plus de la représentation de l'environnement.

## 4. Travail demandé

**Instruction 0.** Proposer des types et structures de données pour représenter une parcelle (nature, état, *etc*) et un environnement.

**Instruction 1.** Écrire une fonction permettant de générer aléatoirement un environnement.

**Instruction 2.** Créer une interface graphique (soit dans le terminal à l'aide de symboles et couleurs, soit à l'aide d'un module `pygame` ou `tkinter`).

**Instruction 3.** Écrire une fonction permettant une génération intelligente de l'environnement : les parcelles sont regroupées aléatoirement en zones homogènes.

**Instruction 4.** Implanter le modèle simple.

**Instruction 5.** Implanter le modèle avec probabilités.

**Instruction 6.** Documenter abondamment le projet par :

- des « *docstrings* »
- des commentaires ponctuels de code (avec `#`) pour expliquer les portions de code ou condition subtiles
- un fichier `README`
- plusieurs *scenari* (c'est-à-dire plusieurs scénarios), correspondant à chacun à un environnement prédéfini, *e.g.* grande plaine américaine, forêts du Nord du Québec, bocage normand, *etc*
- un **rapport de projet**, justifiant les choix effectués sur les structures de données et les algorithmes implantés

## Consignes et évaluation

Les fichiers du code source du projet devront être regroupés et envoyés dans un format de dossier compressé de type `.zip`, ou d'archive compressée au format `.tar.gz` (aucun format `.rar` ne sera accepté). Le dossier compressé devra être nommé au format `prenom1_prenom2_simulateur_incendie.zip` (ni majuscule, ni accent, ni espace), et envoyé par mail à l'adresse `senotprof@gmail.com` **avant le lundi 30/04/2024 à 23h59**, le sujet du mail devant obligatoirement être de la forme : `[NSI] Prénom1 - Prénom2 : projet 4 - simulateur d'incendie`. **Aucun délai supplémentaire ne sera accordé**, la date butoir est immuable, et tout retard sera pénalisé : il convient donc de commencer le projet au plus tôt.

Le projet est à **coder obligatoirement en binôme** (sauf contre-indication ou dérogation). La note sera commune au binôme, sauf cas de travail déséquilibré flagrant.

Il n'est pas interdit (et c'est même conseillé) de discuter du projet entre les différents groupes et de s'échanger

des idées ou de demander un peu d'aide (à commencer par celle de votre enseignant), mais **la rédaction, le style et l'intégralité code doivent être personnels et produits par les membres du groupe** (une fonction ou une portion de code bêtement recopiée ne sera pas utile dans un autre contexte que le programme initial, et surtout ne sera ni comprise, ni assimilée, et sera, en plus des risques de sanctions pour fraude et plagiat, une perte de temps pour tout le monde, élève et enseignant). **Tout plagiat fera l'objet de sanctions avec une tolérance strictement négative.** Une mini-séance de questions par binômes pourra avoir lieu après la remise des projets pour s'assurer que le programme a bien été conçu et implanté par les membres du groupe.

L'évaluation prendra en compte :

- les erreurs de syntaxe : les fichiers .py doivent être directement interprétés sans erreur par la console Python
- **l'originalité du code** (dans le sens de « code personnel et individuel »)
- l'implantation des fonctions imposées et le **respect des consignes** (règles du jeu, cahier des charges, consignes sur le code, les structures de données et les spécifications des fonctions, mais aussi les consignes annexes, comme l'intitulé du mail et le format de fichier, ainsi que bien évidemment le respect des délais)
- la propreté et la lisibilité du code ainsi que tout ce qui facilitera sa compréhension par le correcteur (commentaires du code, noms des variables, commentaires du code, noms des fonctions, commentaires du code, « aération » du code, commentaires du code, etc)
- la correction des fonctions
- la paramétrabilité de la simulation
- la facilité d'utilisation et le réalisme de la simulation produite

## Suivi et rapport de projet

Le suivi de l'évolution et la finalisation du projet sera constituée de :

- un bilan intermédiaire, dont la date sera précisée ultérieurement, et qui consistera en la rédaction d'un mail détaillant l'avancement du projet, ainsi que les fichiers Python en l'état à la date du bilan intermédiaire
- un rapport écrit, de préférence un tapuscrit, d'entre 3 et 5 pages contenant :
  - la répartition détaillée du travail dans le groupe
  - la progression chronologique et les dates approximatives des différentes versions
  - un schéma simple présentant les interfaces et interactions principales des différents modules et fonctions
  - le récit détaillé d'un problème spécifique rencontré lors du déroulement du projet (chronologie, nature et contexte du problème) et de sa gestion et résolution (choix faits et justifications techniques de la solution implantée)
- d'une mini-séance d'au maximum 10 min consistant en quelques questions ciblées à chacun des membres du groupe

## 5. Améliorations (travail optionnel) : lutte contre l'incendie

Des pompiers peuvent intervenir dans l'environnement pour éteindre l'incendie en apparaissant sur une parcelle. À chaque unité de temps, ils abaissent l'intensité du feu de deux niveaux sur les huit parcelles voisines. Une parcelle qui n'a brûlé au maximum que d'une intensité  $i \leq (\text{intensiteMax}/2)$  pourra revenir à son état initial si le feu est éteint, sinon elle se transformera nécessairement en parcelle calcinée. À chaque unité de temps, le pompier peut se déplacer dans un rayon de trois parcelles. Il ne peut pas se positionner sur une parcelle contenant de l'eau ou en feu. La parcelle sur laquelle est le pompier ne peut pas brûler. Un nouveau pompier peut être placé sur le terrain si besoin toutes les 10 unités de temps.

La simulation se termine lorsque la situation n'évolue plus. Calculez ensuite un score en fonction de l'état de l'environnement (les parcelles de plaines, de forêts et de maisons rapportent respectivement 1, 2 et 3 points).

**Instruction 7.** Implanter ces modifications (le positionnement d'un pompier se fera à l'aide de la souris).

**Instruction 8.** Définir une stratégie automatique de lutte contre l'incendie, *i.e.* une intelligence artificielle.

## Autres propositions de fonctionnalités optionnelles

Afin de rendre la simulation encore plus réaliste, de nombreuses améliorations du programme et fonctionnalités optionnelles sont envisageables :

- importer / exporter des cartes au format csv
- afficher uniquement une portion de la carte, *via* une fenêtre que l'on peut déplacer

- renforcer la méthode de lutte contre l'incendie en prenant en compte l'intervention de canadiens (dont l'action est définie sur plusieurs parcelles et est soumise aux aléas du vent) et/ou de tranchées coupe-feu (mises en place avant le début de la simulation)
- ajouter différents types de terrains (types de forêt, zones urbaines différentes, *etc*) avec différentes probabilités de prendre feu
- affiner la génération intelligente du terrain : prise en compte de l'altitude, rivières, *etc*
- améliorer l'interface (ajout de fonctionnalités utilisateur : déclencher plusieurs feux à la souris, les éteindre, *etc*) et les graphismes
- rendre les programmes plus efficaces
- proposer et implémenter des améliorations de la simulation

D'autres fonctionnalités peuvent être proposées et implantées, sous réserve de validation par l'enseignant.

## 6. Annexe : un exemple

On donne comme exemple l'excellentissime et original simulateur implanté en mai 2020 par Célian BUTRÉ, Gabriel LATERRADE et Laura LY (1<sup>ère</sup> NSI, EIB 2019–2020) : l'interface du logiciel en FIGURE 3 et quelques étapes d'un exemple d'exécution en FIGURE 4. En plus de la qualité de la simulation, notons l'originalité et la créativité des très belles icônes de l'interface, conçues et dessinées pour ce projet par Laura Ly.

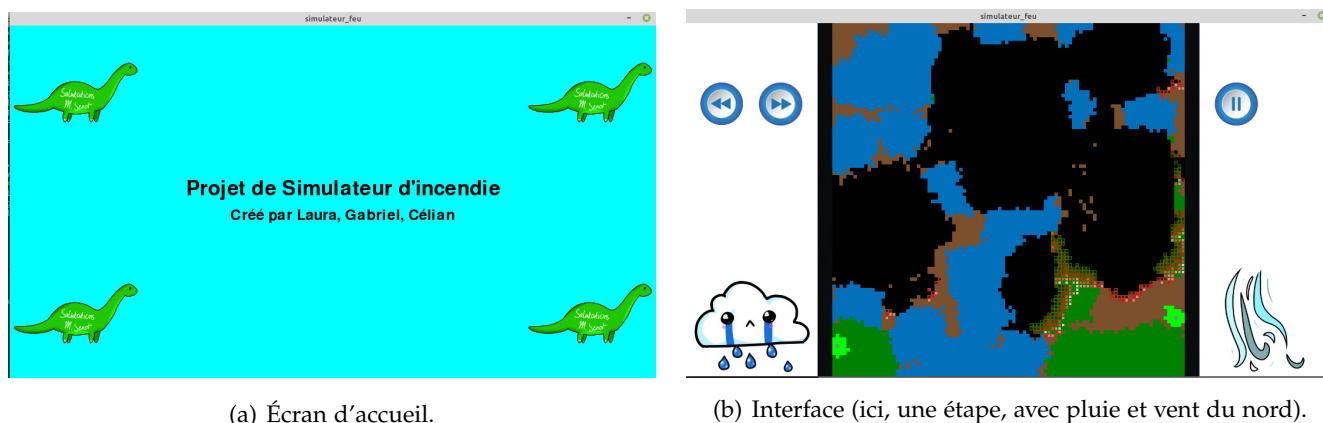


FIGURE 3 – Interface graphique du projet de Célian BUTRÉ, Gabriel LATERRADE et Laura LY (1<sup>ère</sup> NSI, EIB 2019–2020).

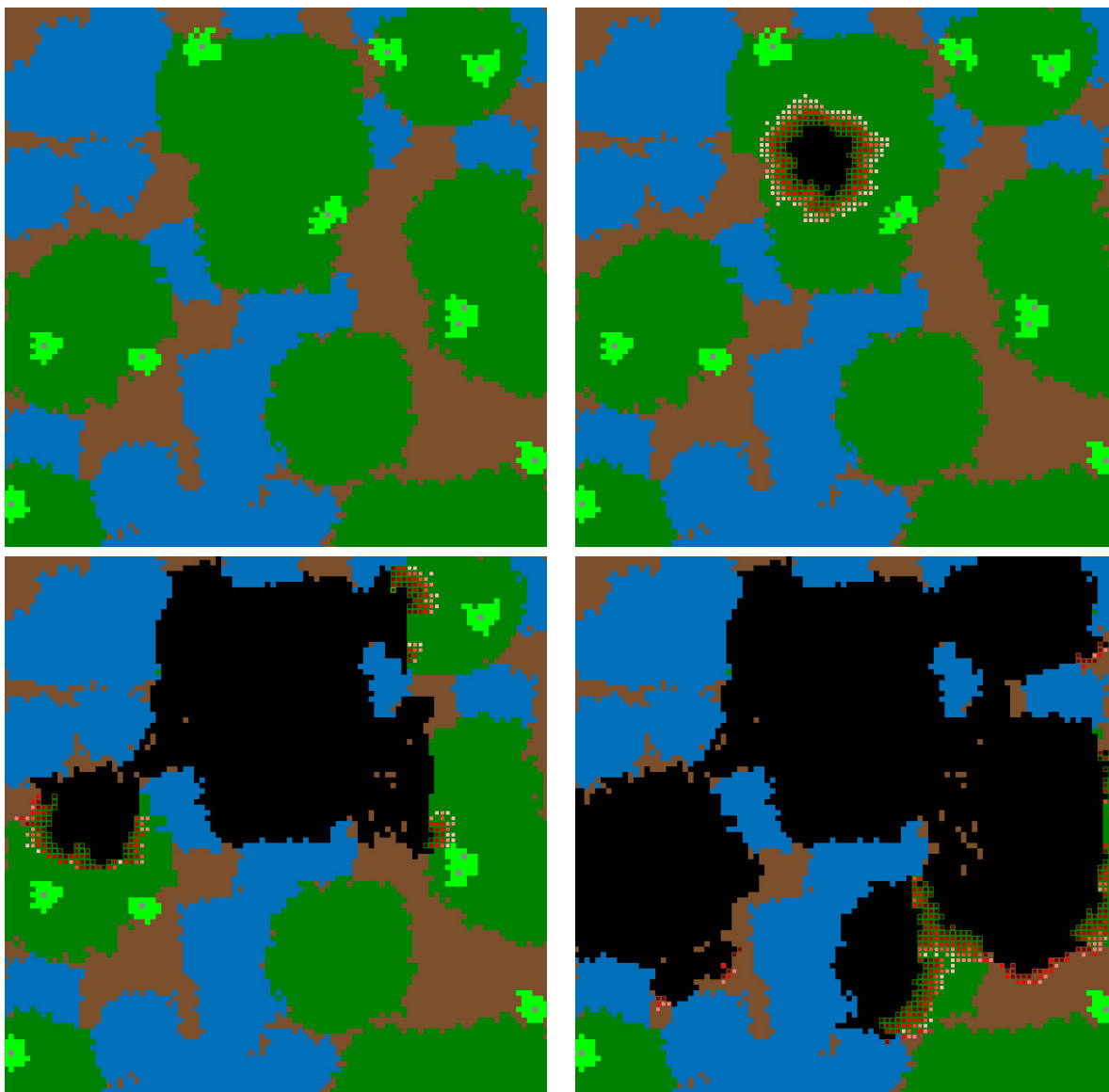


FIGURE 4 – Quelques étapes d’une simulation d’incendie par le programme de Célian BUTRÉ, Gabriel LATERRADE et Laura LY (1<sup>ère</sup> NSI, EIB 2019–2020).