

431 avenue Gaston Berger

13625 Aix-En-Provence

Téléphone 04 42 93 90 43

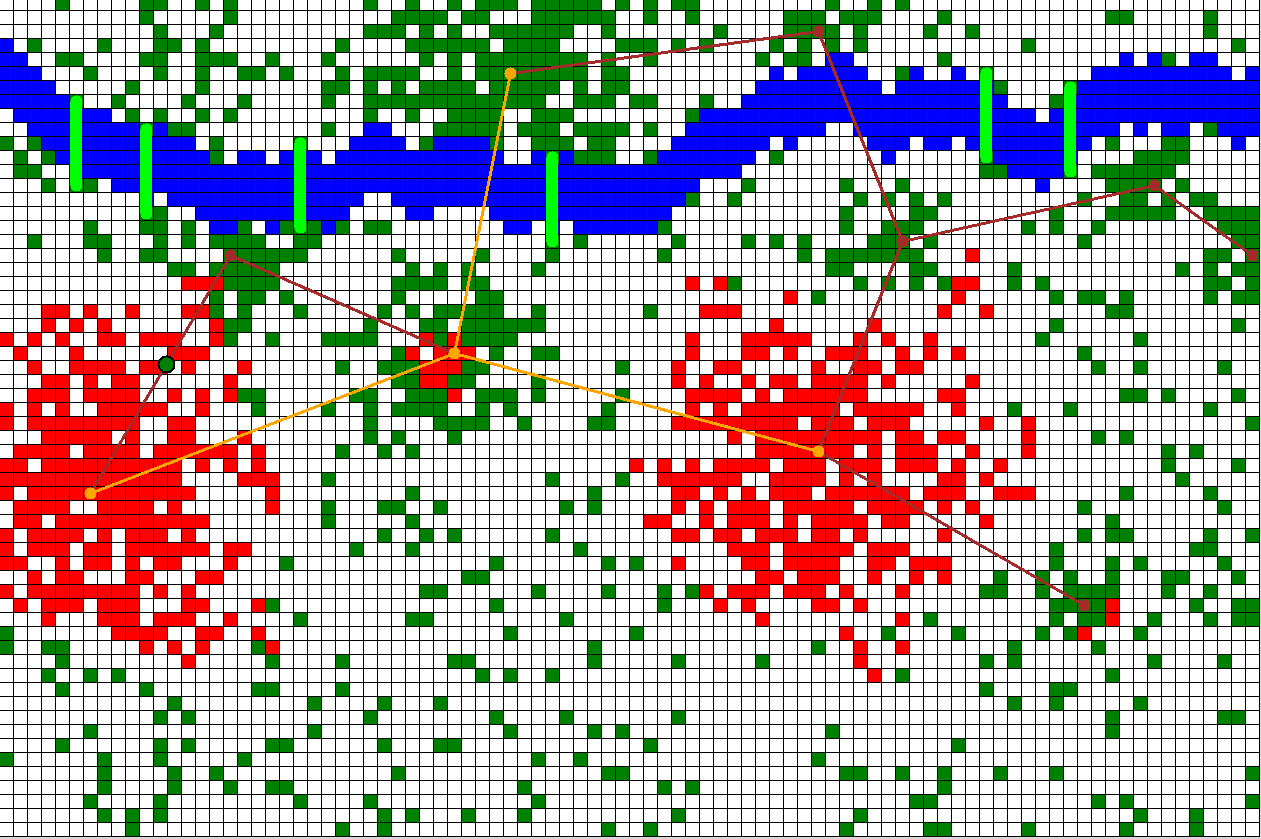
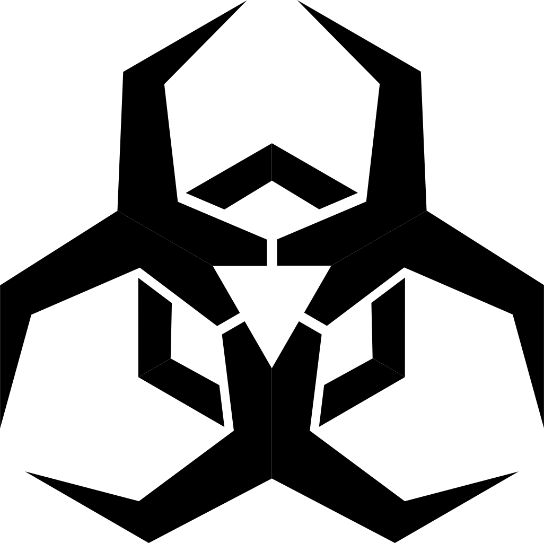
Fax 04 42 93 90 74

[iut-aix-informatique@univ-amu.fr](mailto:iut-aix-informatique@univ-amu.fr)

www.iut.univ-amu.fr

Modélisation Mathématique

Projet automate cellulaire

Dossier final

Aurélien ROBINEAU, Damien PIEDANNA,

Nicolas MEYNIEL, Axel PISANI

Groupe 3 – 2eme année

Enseignent responsable : Patricia GAITAN

Module M3202 – Modélisation mathématique

Mercredi 7 Novembre 2018

Table des matières

[Présentation du projet 3](#_Toc527018504)

[Le code 4](#_Toc527018505)

[Génération de la matrice et des cases 4](#_Toc527018506)

[Outil et source 5](#_Toc527018507)

## Présentation du projet

Nous avons choisi de modéliser, dans le langage de programmation Python, un automate cellulaire.

Chacune de ces cases représentent un groupe de 5 personnes. La globalité de la grille est une zone urbaine générée de manière aléatoire ainsi que tout ce qui compose cette dernière et se faisant infecter par un virus au coure du temps, c’est-à-dire, jour par jour.

Il y a différents types de case :

* les cases saines, donc non infecter par un virus (verte)
* les cases immunisées (vert claire)
* les cases infecter par un virus (rouge)
* les cases mortes (gris)
* les cases dépourvues de population (blanche)
* les cases fleuve (bleu)\*

Ces cases sont dans la globalités regroupées de différente manière :

* zone peuplé
* en village
* en ville
* en métropole

et cela en fonction du nombre de case regroupée.

Pour rendre la contamination plus complexe, un fleuve (case bleu) est généré traversant toute la zone. Il peut être traversé par des lignes ferroviaire (jaune), les routes (marron) et les lignes aérienne (rose) qui relis les villes, villages et métropoles entre eux. Chaque départ et arriver se fait.au centre.

L’initialisation du départ de l’infection se fait avec le clique souris et est nécessaire pour commencer la simulation. Le virus commence par infecter la case courante, au jour 0, avant de progresser vers les cases qui lui sont voisines. Les cases voisines sont les 8 cases autour de la case courante ainsi que les 16 cases entourant ces dernières. Chaque jour passé à moins de 2 cases d’une case infectés, la probabilité de contamination de cette dernière augmente. Cependant, on ne peut pas infecter une case dépourvue de population ou représentant un fleuve.

Le but de cette automate cellulaire est de rendre toute cette zone infectée par le virus, avec le moins de survivant possible.

Pour ce faire nous avons pris en compte :

* l’âge moyen de chaque cellule, constituée d’une population de tout âge (0 à 105 ans)
* le temps en jours passé proche d’une case infectés
* le taux d’infection du virus

### Automate cellulaire

Un automate cellulaire consiste en une grille régulière de « cellules » contenant chacune un « état », choisi parmi un ensemble fini et qui peut évoluer au cours du temps. L'état d'une cellule au temps t+1 est fonction de l'état au temps t d'un nombre fini de cellules appelé son « voisinage ». À chaque nouvelle unité de temps, les mêmes règles sont appliquées simultanément à toutes les cellules de la grille, produisant une nouvelle « génération » de cellules dépendant entièrement de la génération précédente.

## Le code

### Algorithme de Prim

L'algorithme de Prim est un algorithme dit glouton, qui calcule un arbre couvrant minimal dans un graphe connexe valué et non orienté. En d'autres termes, cet algorithme trouve un sous-ensemble d'arêtes formant un arbre sur l'ensemble des sommets du graphe initial, et tel que la somme des poids de ces arêtes soit minimale.

Dans notre projet, chaque sommet est représenté par le centre des villages, villes et métropoles. Afin de créer des liaisons, ici représentés pour les axes routiers, ferroviaires et aériens, nous avons implémenté l’algorithme de Prim et utilisé pour chaque axe. Le poids minimal est la distance entre chaque ville.

## Outil et source

Pour réaliser notre projet d’automate cellulaire, nous avons constamment travaillé en équipe.

Nous avons utilisé les logiciels :

* Sublime Text : open source et qui remplace CoCalc en nous permettant e travailler en local et prend en compte le langage Python (tout comme 44 autre langages de programmation)
* Microsoft Word : pour toute la rédaction du présent dossier.

PROBA

* GenererAgeMoy l 80
* Fleuve 149 – 202
* zoneUrbaine 215-239
* composition de grille 355-407
* genererZoneUrbaine 418 455
* lancerVoyage
* soumettreVirus

MATRICE

* genererDeplacment 664