

Contamination

Rapport de projet C++

Guillaume Roy - ENSAI

06 Avril 2020

1 Fonctionnalités basiques

1.1 Paramétrage de la zone

La zone est principalement paramétrée par un objet **Config**, dont les attributs sont en public et sont, de base, ceux proposés dans le sujet de l'exercice. Ces paramètres sont accessibles à n'importe quel utilisateur : un pointeur vers la configuration est accessible via un getter, et les attributs de cette classe sont tous modifiables.

Néanmoins, dans une expérimentation avec le modèle standard, les interactions avec Zone dans d'autres parties du code seront pour définir sa taille, via le constructeur, ainsi que pour lancer une simulation d'une taille demandée. Dans notre démonstration, ceci se fait dans **main.cpp**.

1.2 Comportement des agents

Les agents ont un comportement uniforme et standard. Tout d'abord, leur génération sur la zone est faite par le biais d'une loi uniforme, qui répartit la population de manière homogène sur la zone. Leur déplacement est ensuite, comme proposé, défini en direction par une loi uniforme, et en vitesse par une loi normale centrée en 1 et d'écart-type 0.3. Les agents se déplacent ensuite à une vitesse constante dans la zone, et rebondissent sur les parois le cas échéant. Leur état est contaminé ou sain, un agent contaminé le restant tout la durée du processus, et un agent sain ne devenant contaminé que sous certaines règles.

1.3 Règles de contamination

La contamination se produit sous deux conditions :

- Les agents doivent être assez proches. La distance est définie par **double contaminationDistance**, actuellement fixée à 1. La contamination ne peut pas se produire si les agents sont trop loin.
- Si la distance est suffisante, un tirage de loi uniforme entre 0.0 et 1.0 se produit. Si le nombre tiré est inférieur à une variable **double probaContamination**, alors l'agent est contaminé. Cette variable fut posée pour symboliser divers éléments aléatoires pouvant affecter la contamination : toux, contact physique etc. Cette variable est actuellement fixée à 0.003

1.4 Affichage

L’affichage se fait par le biais du terminal. Il est effectué dans la méthode **Zone::step**, et montre plusieurs données, relatives à chaque cycle de la pandémie. On montre d’abord les nouveaux contaminés de ce cycle, le nombre total de contaminés, ainsi que finalement l’intégralité des positions des agents, avec une étoile indiquant si un agent est infecté ou non.

Cet affichage ne se produit que si l’argument **bool show** est défini comme **true**. Une autre partie de l’affichage, systématique, est une barre de chargement indiquant l’avancée de la simulation.

2 Fonctionnalités avancées

2.1 Mappage en tronçons

L’algorithme de base proposait une approche heuristique pour la propagation de la maladie : les agents se déplaçaient tous, puis ensuite contaminaient leur entourage en vérifiant tous les agents de la zone. Cette méthode a l’inconvénient, au mieux, de donner $\frac{n(n+1)}{2}$ opérations, et ralentit considérablement la procédure avec chaque agent supplémentaire. La solution mise en place fut de mettre en place un quadrillage sur la zone. Cette dernière est découpée en tronçons (représentés par la classe **Chunk** dans le code), tous des carrés de même taille. Chaque tronçons possède en attribut un vecteur d’agents, qui représente les agents présents dans cette zone. La taille des tronçons peut être définie dans **Config**, bien que leur taille minimale soit forcément de $\frac{\text{contaminationDistance}}{\sqrt{2}}$. Ce système permet à chaque agent de n’avoir à vérifier que sont tronçon, ainsi que les tronçons adjacents : du fait de la taille minimale des tronçons, les autres sont trop loin et aucun agent à l’intérieur ne pourrait être contaminé. Ce système permet la simulation de l’épidémie sur des grandes surfaces en des temps relativement raisonnables.

2.2 Historique de l’épidémie

Si la méthode **Zone::run** est appelé avec la valeur **true** pour l’argument **bool logToFile**, alors la méthode produira un fichier .csv. Ce fichier est nommé de la manière suivante : "log_nbAgents_nombreDeSteps_widthxheight.csv". Il contient un historique de l’épidémie, et permet d’analyser la propagation de la maladie.

