RENIN : Système Multi-agents

Chaque exercice doit faire l'objet d'une sauvegarde dont le nom comportera le numéro de l'exercice : « exo1a.nlogo », puis « exo2a.nlogo »… **Ces fichiers sont à envoyer sous la forme d'un dossier zippé à l'adresse mail suivante : serge.lhomme@u-pec.fr**

1) Appui sur l'existant : analyse de risque

a) A partir du fichier « feu2.nlogo », ajoutez un test à la fin de chaque simulation de propagation de feu permettant de comptabiliser le nombre de scénarios qui engendre la destruction de plus de la moitié de la végétation (des arbres). N'oubliez pas de créer une variable pour cela.
b) Créez un graphique représentant le niveau de risque (ici la proportion des scénarios engendrant la destruction de plus de la moitié de la végétation) en fonction de la densité.
c) Créez un slider permettant à l'utilisateur de fixer le taux de destruction servant de seuil à la quantification du niveau de risque.

Partie exploratoire

Vous allez programmer la diffusion d'un virus au sein d'une population où les personnes ont un comportement étrange. En effet, prises de panique, elles se déplacent de manière aléatoire !

2) Programmer une marche aléatoire

a) A l'aide d'un bouton « Setup » et de sa fonction correspondante, créez 1 personne (un turtles) qui aura une position aléatoire (pour rappel : vous connaissez les commandes « setxy » et « random » ; votre grille comporte 33 pixels en largeur et en hauteur). Laissez la forme définie par défaut.

b) Vous allez maintenant chercher à déplacer cette personne à l'aide d'un bouton « Go » et de sa fonction correspondante. Pour cela, ce n'est pas très compliqué vous pouvez lancer une commande du type : « ask turtles [right 0 forward 1] ».

c) Si vous testez différentes valeurs de déplacement (comme par exemple « ask turtles [right 90 forward 1] ») vous remarquez que « right » permet de définir la direction du déplacement (en fonction de l'orientation initiale du turtles) et que « forward » correspond à la distance du déplacement. A partir de ces informations, programmez un déplacement aléatoire qui s'appuie sur une définition aléatoire de l'angle et une distance associée égale à 1.

d) Pour obtenir la trace de cette marche, associez une couleur rouge aux « patches » parcourus par votre « turtles ». Ensuite, éditez le bouton « go » afin que les itérations se fassent de manière automatique et programmez l'arrêt du programme pour 1000 itérations (rappel : il est possible de créer une variable globale « iteration »).

3) Programmer un processus de contagion

Programmons maintenant le processus de contagion. Pour cela, repartez d'un fichier vide. On réintroduira la marche aléatoire un peu plus tard.

a) A l'aide d'un bouton « Setup » et de sa fonction correspondante, créez 1000 personnes (turtles) de couleur verte qui auront une position initiale aléatoire (ce n'est pas grave si elles se superposent). Laissez la forme définie par défaut.

b) Toujours à l'aide du bouton « Setup », créez aussi 100 personnes de couleur rouge qui auront une position aléatoire (ce n'est pas grave si elles se superposent).

c) En vert, nous avons des « personnes saines » et en rouge nous avons des « personnes infectées » par un virus. Programmez l'infection des personnes saines qui sont en contact direct avec les turtles rouges à l'aide d'un bouton « Go » et de sa fonction

correspondante. Pour cela, demandez aux personnes infectées (qui ont donc la propriété d'être rouges), d'infecter (c'est-à-dire de mettre en rouge) les personnes qui se trouvent au même endroit (rappel : la commande « turtles-here » vous sera utile pour identifier les turtles qui se situent au même endroit).

d) A la suite de cette infection, toutes les personnes (infectées et saines) se déplacent aléatoirement (c'est-à-dire que la définition de l'angle de déplacement est aléatoire et la distance associée est toujours égale à 1 comme à la question 2c).

e) Produisez le graphique (le plot) qui représente le nombre de personnes infectées (le nombre de turtles rouges) au cours du temps.

f) Editez le bouton « Go » afin que les itérations se déclenchent de manière automatique et programmez l'arrêt du programme dès qu'il n'y a plus de personnes saines (c'est-à-dire plus de turtles verts).

4) Etudier un processus de contagion

Au bout d'un certain nombre d'itérations, toutes les personnes sont infectées. Ce qui n'est pas très réaliste et intéressant à étudier. Ainsi, on aura beau faire varier les conditions initiales (le nombre de personnes saines ou le nombre de personnes infectées), on aboutit toujours au même résultat. Pour complexifier le processus de contagion, il suffit de rajouter quelque chose de très simple : les personnes infectées par ce virus meurent systématiquement.

a) A partir du fichier « exo3d.nlogo », programmez la mort des personnes infectées après le processus de transmission de l'infection.

b) Produisez un graphique qui représente le nombre de personnes infectées au cours du temps.

c) Désormais, compte tenu des conditions initiales, les personnes infectées ne sont pas assez nombreuses pour contaminer l'ensemble de la population. Créez alors un slider qui fera varier le nombre initial de personnes infectées (de 1 à 10000).

Compte-tenu de la mort rapide des personnes infectées, il faut initialement un grand nombre de personnes infectées pour tuer toute la population. En effet, le virus est paradoxalement trop virulent pour se répandre au sein d'une population saine même nombreuse. Nous souhaitons donc faire durer la période de contamination. Pour cela, il faut créer une variable définissant la durée de vie après une infection.

d) La commande « turtles-own [dureedevie] » placée avant la fonction « Setup » permet de définir cette variable. Retirez la commande « die » mise précédemment. Dans la fonction « Setup », lorsque vous créez vos personnes, initialisez la variable « dureedevie » à 5. Puis, dans la fonction « Go », à chaque fois qu'une personne infectée est sollicitée pour infecter d'autres personnes, retirez lui un point de vie. Au début de votre fonction « Go », demandez aux personnes de mourir si elles ont une variable «dureedevie» inférieure à 1.

e) Redéfinissez le slider du nombre initial de personnes infectées avec des valeurs comprises entre 0 et 1000 et créez un autre slider qui permet de faire varier la durée de vie après une infection pour des valeurs comprises entre 1 et 10. Définissez l'arrêt de votre fonction dès qu'il n'y a plus de personnes infectées vivantes.

Ca y est vous avez un modèle intéressant à étudier… Vous pourrez chercher à produire des graphiques qui représentent le taux de personnes mortes en fonction de la durée de vie après l'infection et/ou du nombre initial de personnes infectées et/ou du nombre initial de personnes saines…