

**SIMULATION DE POPULATION: LES SHMOKELS**

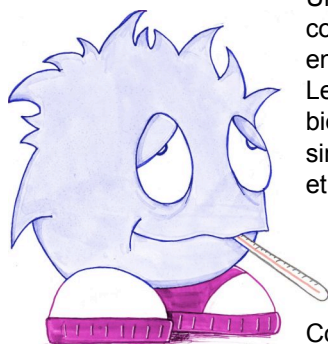
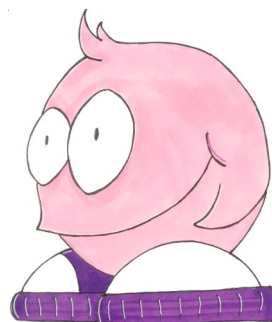
Dirk Amadori, Arnaud Germis, Nadia Kabbali, Edouard Klopfert, François Petit
Département d'informatique

Ce programme informatique de simulation a pour but de dégager la ou les façons dont réagit une population (les Shmokels) face à certains types de problèmes, tels que la recherche de nourriture et la transmission des maladies virales, génétiques et sexuellement transmissibles, dans différents environnements.

Pour cela, nous retenons pour les individus et les environnements, un certain nombre de caractéristiques qui semblent jouer un rôle dans les situations simulées. Bien sûr, cette modélisation de la réalité reste limitée, mais elle pourrait permettre de comprendre sur quels facteurs agir dans des cas réels et quelles sont les causes et les effets de la propagation des maladies ou des déplacements de populations pour mieux les gérer.

Ce type de programme pourrait profiter à tout le monde (Qui a envie de tomber malade ?!), mais surtout aux Etats, qui pourraient prendre des mesures par rapport aux causes de propagation pour limiter l'impact d'une épidémie, voire d'une pandémie.

Aussi, la modélisation ne se limite pas à l'humain, étant donné que les caractéristiques des individus restent basiques. Elle pourrait donc servir à comprendre certaines réactions animales, ou même la façon dont une épizootie se répand, ce qui pourrait profiter, par exemple, aux biologistes ou aux sociologues.



Un algorithme (suite d'instructions, informatiques dans notre cas) simule le comportement de chaque Shmokels en fonction de ses propriétés et de son environnement.

Le comportement de la simulation ainsi que celui de ses habitants dépend, bien évidemment, de la précision des données des propriétés retenues pour simuler le comportement des Shmokels. Au plus les données sont précises et l'algorithme évolué, au plus la simulation est proche de la réalité.

Comment réalise-t-on un tel programme?

On part d'une version de base très simple, fonctionnelle, que l'on enrichit par la suite, en ajoutant des propriétés, ainsi qu'en augmentant la précision des algorithmes de simulation utilisés.



SIMULATION DE POPULATION: LES SHMOKELS

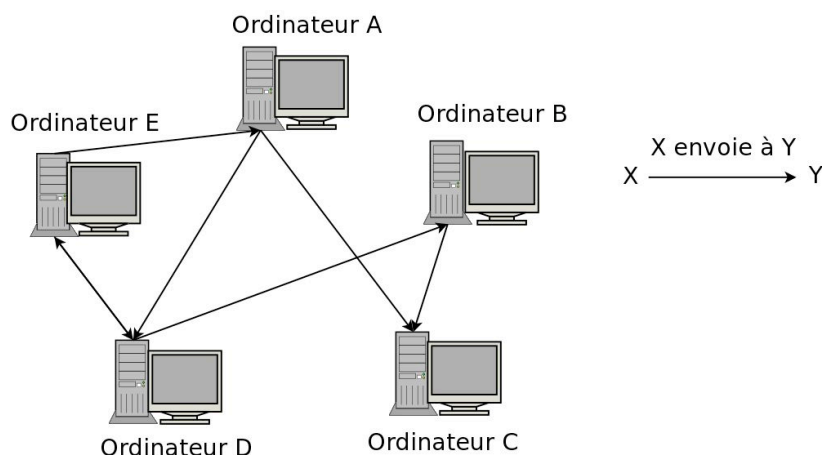
Dirk Amadori, Arnaud Germis, Nadia Kabbali, Edouard Klopfert, François Petit
Département d'informatique

Afin de ne pas choquer les visiteurs, nous avons simulé une population imaginaire de Shmokels, qui ont des caractéristiques communes à toutes les espèces vivantes.

Ces Shmokels se nourrissent, se reproduisent, se déplacent dans leur environnement... mais ils peuvent également changer d'environnement, s'ils en ont envie.

Nous avons utilisé pour ce faire un réseau d'ordinateurs, où chaque ordinateur fait tourner un environnement et est connecté (logiquement) à d'autres, nous avons donc plusieurs environnements interconnectés.

Voici la représentation d'un tel réseau :

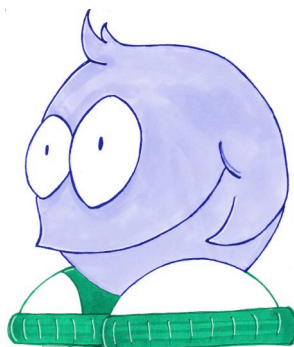


Ainsi, les Shmokels de l'ordinateur A, ont le choix entre migrer vers l'ordinateur D ou C.

Ceux de B, peuvent migrer seulement vers l'ordinateur C.

Quant aux Shmokels de D, ils peuvent migrer vers B ou vers E...

Les Shmokels décident de migrer vers un autre environnement en fonction de leur état et de facteurs aléatoires.





SIMULATION DE POPULATION: LES SHMOKELS

Dirk Amadori, Arnaud Germis, Nadia Kabbali, Edouard Klopfert, François Petit
Département d'informatique

Le programme est majoritairement écrit en C++ (un langage de programmation orienté objet) mais la partie réseau est écrite en C (l'ancêtre de C++).

La programmation orientée objet permet de décomposer le programme en différents éléments plus simples et se prête bien à un développement itératif (par étapes successives).

L'environnement est représenté dans notre programme sous la forme d'une matrice (tableau). Chaque élément de la matrice peut contenir un Shmokels ou de la nourriture (ou tout autre élément positionnable). Les coordonnées de la matrice donnent une position à chaque élément dans l'environnement.

Les ordinateurs sont connectés physiquement entre eux (par un câble réseau par exemple). Les programmes tournant sur les différentes machines doivent ensuite faire «connaissance», à l'aide de l'adresse IP de leur machine respective. Le protocole utilisé est le protocole TCP/IP.

Notre programme comprend une interface graphique, ainsi qu'une interface texte, qui permettent l'affichage du programme, ainsi que l'interaction de l'utilisateur avec le programme.

