Master IA

Systèmes mutli-agents

TP2

Ex 1 : Configuration d'un projet « P5 ». Vous pouvez suivre le tutoriel sur cv.bdafflon.eu

1. Pull ou Cloner le dépôt https://github.com/bdafflon/p5
2. Configurer le projet pour utiliser un virtual environment
3. Installer pygame si nécessaire
4. Lancer l'exemple « boids » pour vérifier la configuration

Ex 1.Bis

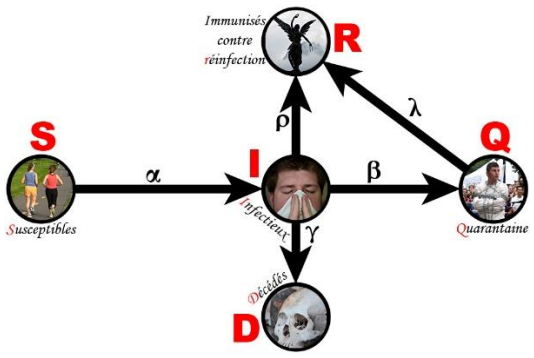
Si vous préférez utiliser Javascript, vous pouvez utiliser P5.js qui fonctionne de la même manière.

Ex 2 : Modèles compartimentaux en épidémiologie

Les modèles mathématiques de maladies infectieuses, d'abord outils purement théoriques, ont commencé à être mis en pratique avec le problème du SIDA dans les années 1980. Lors de la pandémie Covid 19, les modélisations mathématiques ont connu un essor lors de la prise de décision relatives aux politiques de santé publique et a également contribué à l'épidémiosurveillance de la maladie.

Principes fondamentaux : Un modèle épidémiologique se fonde sur deux concepts : les compartiments et les règles. Les compartiments divisent la population en divers états possibles par rapport à la maladie. Par exemple, William Ogilvy Kermack et Anderson Gray McKendrick proposèrent un modèle fondateur dans lequel la population était divisée entre les individus susceptibles de contracter la maladie (compartiment S), et les individus infectieux (compartiment I).

Les règles spécifient la proportion des individus passant d'un compartiment à un autre. Ainsi, dans un cas à deux compartiments, il existe une proportion p(S → I) d'individus sains devenant infectés et, selon les maladies, il peut aussi exister une proportion p (I → S) d'individus infectieux étant guéris. L'acronyme utilisé pour un modèle est généralement fondé sur l'ordre de ses règles. Dans le modèle S I R un individu est initialement sain (S), peut devenir infecté (I) puis être guéri (R). On peut représenter ces règles par le schéma suivant :



Vous trouvez un exemple sur la page du CNRS : http://images.math.cnrs.fr/Modelisation-d-une-epidemie-partie-1.html

1. Nous allons coder dans le dossier « sma », le fichier « main.py » va nous servir d'environnement.
2. Ajouter les fichiers « agent.py », « body.py », « fustrum.py » et « item.py » conformément au diagramme UML du TD.
3. (1 pts) Ajouter, pour chaque agent, une propriété « statut » indiquant dans quel compartiment de la population il se trouve.
4. (1 pts) Ajouter une méthode « def show(self) : » dessinant un agent tenant compte du statut pour la couleur.
5. (2 pts) Ajouter un comportement « déplacement aléatoire » pour les agents
6. (3 pts) Coder les fonctions computePerception, computeDecision et applyDecision du fichier « main.py »
7. (2 pts) Ajouter un fichier « epidemie.py » contenant un dictionnaire regroupant les paramètres d'une épidémie :
    a. Durée d'incubation
    b. Durée avant contagions
    c. Pourcentage de contagion (combien de chance un agent S a-t-il de passer I au contact d'un autre agent I)
    d. Durée avant décès
    e. Pourcentage de mortalité
    f. Distance mini de contagion
8. (3 pts) Ajouter une méthode « update » dans la classe « body » pour prendre en compte ces paramètre
9. (1 pts) Ajouter un déclencheur d'épidémie : en cliquant avec la souris, l'agent le plus proche devient automatiquement « I ».
10. (2 pts) Ajouter un comportement de groupe : un agent peut appartenir à un petit groupe (une fratrie). Le groupe se déplace ensemble avec une interaction basée sur une loi « masse-ressort »
11. (2 pts) Ajouter un comportement « malade rationnel » qui va chercher à s'isoler et mettre un masque. Le masque diminue par deux les statistiques de contamination. Un agent contagieux a 70% de chance d'avoir ce comportement.
12. (1 pts) Afficher dans la console le pourcentage de la population S,I,R et D
13. (1 pts) Utiliser matplotlib et threading pour afficher un graphique en temps réel des populations.