

<p style="text-align: center;">IA04 - Printemps 2014 Examen final 2 heures <i>Documents interdits</i></p>
--

Exercice 1 (8 pts)

1. Donner une définition de la plateforme WADE.
2. L'environnement d'un agent peut-être accessible ou non accessible. Donner trois autres caractéristiques que peut avoir cet environnement.
3. Comment pourrait-on définir un Middleware Orienté Message (MOM) ?
4. Deux entreprises E1, et E2 n'ont quasiment aucune concurrence sur la vente d'un produit. E1 fait un bénéfice de 1 et E2 fait un bénéfice de 5. Si E2 fait une campagne de publicité et E1 ne fait rien, il n'y a pas de variation de bénéfice. Par contre, si E1 lance une campagne de publicité et que E2 ne réagit pas les bénéfices sont de 2 pour E1 et de 3 pour E2. En revanche, si E2 réagit en lançant sa propre campagne, les bénéfices deviendront de 0 pour E1 et de 2 pour E2.
 - a. Donner la matrice des gains de ces stratégies (ne rien changer et faire de la publicité). Montrer qu'il y a deux équilibres de Nash.
 - b. Pourquoi vraisemblablement E2 ne lancera pas de campagne de publicité ? Quelles sont alors les stratégies des deux entreprises.
5. Citer les quatre méthodes REST principales. Quelle est leur signification ?
On veut pouvoir créer, modifier et obtenir la description d'un document. Proposer un adressage REST pour ceci.

Exercice 2 (6 pts)

Un environnement multi-agents de simulation comporte une grille carrée composée de cellules qui peuvent contenir plusieurs éléments. On considère un agent Leader qui se déplace sur la grille, à une vitesse régulière d'un pas par itération, en laissant une trace de 10 unités sur chaque cellule sur laquelle il passe. La trace s'évapore d'une unité à chaque itération. On considère d'autres agents capables de repérer la trace et qui doivent essayer de la suivre une fois repérée. Leur distance de perception est de 1 cellule dans toutes les directions et leur possibilité maximale de déplacement est également d'un pas par unité de temps. Les agents Leader et suiveurs sont placés initialement de façon aléatoire.

Question :

1. Donner les types d'entités impliquées dans la simulation, leur rôle principal et leurs attributs.
2. Donner l'algorithme suivi par chaque entité à chaque étape de la simulation. On pourra simplement décrire par une phrase ce que réalisent les fonctions de bas niveaux utilisées sans entrer dans les détails.

A faire sur une copie séparée.

Exercice 3 (6 pts)

On désire mettre en place un système de gestion de flux sur la piste d'atterrissage d'un aéroport. On décide de traiter les différentes informations à l'aide d'un système multi-agents basé sur une plateforme JADE dont les agents sont les suivants :

- RadarAgent : repère les avions en approche.
- FlowDirectorAgent a deux missions :
 - o Etablir le planning des atterrissages en fonction des données transmises par les avions et détecter les éventuels encombrements (deux avions ne peuvent atterrir dans un délai de temps trop court).
 - o Assigner une nouvelle heure d'atterrissage aux avions en cas d'encombrement. Le calcul d'une nouvelle heure d'atterrissage est basée sur un algorithme de type A* et doit prendre en compte, pour chaque avion, sa réserve en carburant, ses vitesses minimale et maximale d'atterrissage ainsi que l'intervalle de temps à respecter entre deux atterrissages qui dépend du type d'avion (un gros avion engendre plus de perturbations à l'atterrissage qu'un petit avion).
- AircraftAgent (un AircraftAgent par avion) a deux missions :
 - o Prédire : estimer son heure d'atterrissage.
 - o Planifier : recevoir les directives du FlowDirectorAgent et établir un plan pour les atteindre (ex : réduire la vitesse OU changer de trajectoire).

La séquence d'actions lorsqu'un avion est en approche est la suivante :

1. RadarAgent détecte l'avion et signale au AircraftAgent qu'il entre dans la zone de contrôle de l'aéroport.
2. AircraftAgent transmet son heure estimée d'arrivée au FlowDirectorAgent, ainsi que toutes les informations nécessaires au calcul d'une nouvelle heure d'atterrissage.
3. FlowDirectorAgent établit le planning des atterrissages et si nécessaire, assigne une nouvelle heure d'atterrissage à AircraftAgent.
4. Si AircraftAgent se voit assigner une nouvelle heure d'atterrissage, il planifie les mesures à prendre pour atteindre cet objectif et confirme (ou infirme) au FlowDirectorAgent qu'il respectera (ou pas) la nouvelle heure d'atterrissage.

Questions

1. Quelles informations AircraftAgent doit-il transmettre au FlowDirectorAgent lorsqu'il approche de l'aéroport ? Donner un exemple de contenu de message (format JSON).
2. Etablir un diagramme de messages illustrant la séquence d'échanges ci-dessus. Préciser pour chaque message ses attributs (performatif, contenu, etc.).
3. Décrire l'agent FlowDirectorAgent en termes de behaviours JADE. Pour chaque behaviour, donner les informations suivantes :
 - Type et rôle du behaviour (en une phrase, on ne demande pas de détailler le fonctionnement).
 - Moment où le behaviour est ajouté.
 - Critère de terminaison des behaviours (s'ils se terminent).
 - Possibilité d'avoir plusieurs behaviours de ce type.
4. Décrire en pseudo code le behaviour qui permet de calculer une nouvelle heure d'atterrissage pour les avions en cas d'encombrements. (La fonction *replanning()* basée sur l'algorithme A* n'est pas à écrire).