

Session : .....1

Durée de l'épreuve : 2 heures

Date : 11 / 01 / 2017

Documents autorisés : aucun

Licence ☐ Master M1 Informatique

Mention : ..... INFORMATIQUE

Matériels autorisés : aucun

Parcours : ..... IMAGINA, DCOL, AIGLE

Libellé + Code de l'UE : **HMIN 108 – Programmation par agents**

Responsable : J. Ferber

## SUJET

### Exercice 1 : Questions générales

- 1) Que signifie le terme '**Reporter**' en NetLogo
- 2) En NetLogo, donnez l'expression qui retourne la tortue de couleur '**green**' qui la plus proche de la tortue courante.
- 3) Que retourne l'expression **patch 0 0** en Netlogo (soyez précis)
- 4) Que fait la fonction **uphill** en NetLogo. Donnez un exemple d'utilisation de cette fonction.
- 5) En NetLogo, donnez le code permettant de colorier en bleu tous les patches qui se trouvent à un rayon de 5 à partir de la tortue courante.
- 6) En NetLogo que fait la fonction **scale-color** et quels sont ses arguments. Donnez un exemple de code utilisant cette fonction.
- 7) En NetLogo, que fait la fonction **in-cone** et quels sont ses arguments. Donnez un exemple de code utilisant cette fonction.
- 8) Quelle est la différence entre un **agent** du point de vue des systèmes multi-agent, et ce qu'on appelle **agent** en NetLogo.
- 9) Que fait la fonction **turtles-here** en Netlogo.
- 10) Qu'est ce qu'un 'champ de potentiel' et un 'gradient' ? Comment utilise-t-on ces concepts dans des systèmes multi-agents? Soyez précis !!

### Exercice 2 : Architectures

1. Quelle est la différence entre une FSM hiérarchique et une FSM à réflexe.
2. Donnez le diagramme et un exemple de FSM hiérarchique à réflexe.
3. Imaginez dans un jeu video un type de personnage, un **Péon**, dont l'action principale consiste à aller chercher du bois et à le rapporter chez lui (on suppose qu'il y a la tâche déjà écrite **aller-chercher** et **rapporter**).
4. Ecrire en NetLogo le code générique d'une machine FSM hiérarchique à réflexe.

Mais si un **Chef de guerre** vient le solliciter, il enfle une armure et il le suit à la guerre. Or aller à la guerre est une activité complexe, qui consiste à

- **Suivre-le-chef**
  - **Attaquer**, s'il l'on reçoit l'ordre du chef
  - **Défendre** si un ennemi attaque
  - **Eviter** les projectiles qui circulent
5. Dessiner le diagramme d'un **Péon** sous forme de FSM hiérarchique à réflexe. On ne demande pas de décrire chacune des tâches mais simplement l'organisation générale. Soyez précis et clair.



## Exercice 4 : Warbot

On veut définir un protocole de messages dans lequel 1) un **Explorer** envoie un message à sa propre base indiquant l'emplacement d'une base ennemie lorsqu'il en a détecté une. On supposera qu'il n'y a qu'une base dans chacune des équipes.

Ensuite 2) la base, lorsqu'elle reçoit cette information fait un appel d'offre pour trouver 1 **Explorer**, 2 **Heavy** et 2 **RocketLauncher** pour attaquer la base ennemie (et uniquement ce nombre, on ne veut pas plus que ce qui est demandé). On suppose que les unités, lorsqu'elles répondent font un engagement fort (elles ne sont plus disponibles pour un autre appel d'offre pendant un temps **k**, **k** représentant un nombre de cycle déterminé à l'avance). Lorsque la base a recruté ses troupes, elle leur envoie l'ordre d'attaquer la base ennemie avec la position de cette dernière. Si au bout de **k** unité de temps, elle n'a pas recruté les unités demandées, elle arrête cette attaque.

1. Ecrire a) le diagramme de séquence du protocole de l'**Explorer** vis à vis de la base, et b) le diagramme de séquence du protocole qui relie la base et les autres unités avant de lancer l'attaque.
2. Ecrire le code (en Python ou Java) du comportement de l'**Explorer**, ainsi que la partie gestion des messages du comportement de la base.

**Remarque:** Vous trouverez ci-joint un rappel des fonctions de communications dans Warbot. D'une manière générale Si vous faites une erreur concernant le nom d'une primitive Warbot ce n'est pas compté. L'important c'est la logique de ce que vous écrivez. Ecrire le code de la manière la plus claire et la plus lisible possible. La clarté du code est prise en compte dans la notation.

### Warbot : Gestion des messages

- **sendMessage** (int idAgent, String message, String ... content) : ReturnCode → Envoie un message à l'agent d'id iAgent, avec le sujet du message et le contenu.
- **reply**(WarMessage warMessage, String message, String ... content) : ReturnCode → Envoie un message de réponse au message passé en argument.
- **broadcastMessageToAll**(String message, String ... content) (int idAgent, String message, String ... content) : ReturnCode → Envoie un message à tous les agents de l'équipe.
- **broadcastMessageToAgentType**(WarAgentType agentType, String message, String ... content) : ReturnCode → Envoie un message à tous les agents (alliés bien sûr) de type WarAgentType.
- **broadcastMessageToGroup** (groupName, message, content) → Envoie un message aux agents du groupe groupName.
- **broadcastMessage** (groupName, roleName, message, content) → Envoie un message aux agents du groupe groupName, ayant pour rôle roleName.