

Session :1

Date : 13 / 01 / 2016

Licence ☐ Master M1 Informatique

Mention :..... INFORMATIQUE

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

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

Durée de l'épreuve : 2 heures

Documents autorisés : aucun

Matériels autorisés : aucun

SUJET

Exercice 1 : Questions générales

- Quelle est la différence entre un agent cognitif et un agent réactif. Donnez un exemple d'agent réactif et d'agent cognitif.
- Que fait la fonction **max-one-of** en NetLogo. Donnez sa syntaxe et un exemple d'utilisation.
- Que signifie l'erreur suivante en NetLogo liée à la définition d'un bouton. Que faut-il faire pour l'éviter ?

Agent(s) turtles ☐ Forever

☐ Disable until ticks start

Commands

 You can't use GO in a turtle context, because GO is observer-only.

go

Display name

Action key

Cancel

OK

- Que signifie le terme 'Reporter' en NetLogo
- Quelle est la différence entre un **agent** du point de vue des systèmes multi-agent, et ce qu'on appelle **agent** en NetLogo.
- En Netlogo, donnez l'instruction qui permet d'aller vers le patch de coordonnées **px=10,py=10**.
- Que fait la fonction **turtles-here** en Netlogo.
- A quoi sert la case à cocher 'forever' dans un bouton de NetLogo. Que se passe-t-il quand on la coche ?

Agent(s) turtles ☐ Forever

☐ Disable until ticks start

Commands

- Donnez les deux manières (les deux codes) permettant à un agent d'aller vers le patch voisin ayant la plus grande valeur pour la variable **bidule** (suivi de gradients).
- Que retourne l'expression **one-of neighbors with [energy = 100 and taille > 50]** en Netlogo? (soyez précis !)

- k) Donnez le code en NetLogo qui permet de retourner la **reine** la plus proche de l'abeille courante (On suppose que la définition de Reine et d'Abeille a déjà été donnée).
- l) On veut diffuser en Netlogo un signal de nom "parfum" qui se diffuse avec un taux **taux-diffusion** et qui s'évapore avec un taux d'évaporation **taux-evap** à chaque tour. Donnez le code permettant sa création, la gestion de sa diffusion et de son évaporation (on supposera que la procédure globale appelée à chaque tour s'appelle **to go**).
- m) Qu'est ce qu'un 'champ de potentiel' et un 'gradient' ? Comment utilise-t-on ces concepts dans des systèmes multi-agents? Soyez précis !!
- n) Dans Warbot, donnez le diagramme de séquence de communications d'un robot explorateur qui recrute exactement n lanceurs de missiles pour lancer une attaque contre une base ennemie. S'il ne reçoit pas les n réponses positives de lanceurs, il abandonne au bout d'un certain temps (time limit).
- o) Dans Warbot, comment un *Explorer* peut-il communiquer à tous les lanceurs de missiles la position relative (en angle et distance) d'une base ennemie qu'il vient d'apercevoir? Donnez le code de l'Explorer.

Exercice 2 : NetLogo

On veut représenter de l'herbe qui croît. Chaque patch représente une parcelle d'herbe. La variable **taille** représente la hauteur de l'herbe. Initialement, au temps $t = 0$, chaque parcelle a une hauteur initialisée avec un nombre aléatoire compris entre 0 et **taille-max**. On supposera que **taille-max** est représentée sous la forme d'une variable globale.

A chaque tick l'herbe a une probabilité de **proba-croissance** de croître de **taux-croissance**, ces deux paramètres étant définis sous la forme de sliders. On supposera que **proba-croissance** est une variable en nombre flottant exprimée dans l'intervalle [0 1] et que **taux-croissance** est une variable entière.

Si l'herbe atteint **taille-max**, alors elle ne croît plus.

Ecrire l'ensemble du code en NetLogo permettant d'implémenter cette croissance d'herbe.

Exercice 3 : Architectures

1. Quelle est la différence entre une FSM et une FSM hiérarchique ?
2. Donnez le code générique en NetLogo permettant de programmer une FSM et donnez aussi celui permettant de programmer une FSM hiérarchique.
3. Que permettent les « réflexes » dans une FSM ? Est ce qu'il est possible de représenter des réflexes dans une FSM Pure, et comment ? (donnez un tout petit exemple).
4. On suppose qu'un agent puisse passer dans les états **Explorer, Revenir à la Base, Prendre de la nourriture, Décharger**. Il rentre à la base quand son sac est plein et repart quand il a déchargé. Donnez le diagramme en FSM du comportement d'un tel agent
5. On suppose maintenant qu'il doit aussi éviter les obstacles. Comment s'exprime ce nouveau comportement avec une FSM à réflexe, une FSM-hiérarchique et une FSM pure (sans réflexe). Donnez les diagrammes correspondant à ces trois possibilités. Y aura-t-il une différence de comportement et lequel ?
6. Donnez le diagramme correspondant à l'architecture de subsomption décrivant le comportement de cet agent (avec l'évitement des obstacles).
7. On décide d'implémenter cette activité en Warbot, pour les explorateurs en utilisant l'architecture de subsomption. Donnez le code de la fonction **action()** permettant d'implémenter cette activité (en Java ou Python) dans l'explorateur.

Rappel : les actions 'take', 'drop' pour prendre et déposer de la nourriture. Il ne vous est pas demandé de vous souvenir précisément de chaque commande Warbot. C'est la logique générale qui est importante.