

Examen programmation par agents

M1 informatique – FMIN207 – Janvier 2014

Responsables J.Ferber – Durée 2 h - Documents interdits.

Exercice 1 : Questions générales

- a) Qu'est ce que 'of' en NetLogo. Donnez un exemple de son utilisation.
- b) Que font les fonctions 'any?' et 'empty?' en NetLogo et quelle est leur différence?
- c) Quelle est la syntaxe et la définition (sa sémantique si vous préférez) de la commande **ask** en NetLogo. Donnez un exemple d'utilisation de cette commande tiré de vos TP.
- d) Donnez la syntaxe et la définition (sa sémantique si vous préférez) de **breed** en NetLogo. Donnez un exemple d'utilisation.
- e) Qu'est ce qu'un champ de potentiel dans les systèmes multi-agents. Donnez un exemple d'utilisation des champs de potentiels venant des TP.
- f) Donnez au moins trois types d'architectures réactives
- g) En NetLogo, écrire le code de la fonction **fibonacci(n)** donnée récursivement ainsi :

```
fibonacci(0) = 0
fibonacci(1) = 1
fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)
```
- h) Que retourne l'expression **one-of neighbors with [pcolor = green and hauteur > 50]** en NetLogo? (soyez précis !)
- i) Quelle est la différence entre FSM (machine à états finis) et subsomption. Donner un exemple de situation simple et le coder sous forme de FSM et de Subsomption en NetLogo.
- j) On veut diffuser en NetLogo un signal de nom "parfum" qui s'évapore de 4% à chaque tour. Donnez le code permettant sa création, la gestion de sa diffusion et son évaporation (on supposera que la procédure globale appelée à chaque tour s'appelle **to go**)
- k) 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*.
- l) Dans Warbot, écrire le code de l'expression permettant à un *Explorer* de récupérer de la nourriture.

Exercice 2 : programmation en NetLogo

- a) On désire qu'un agent de type **Suiveur** suive un autre agent de type **Leader** dès qu'il l'aperçoit (dans un rayon de perception **dist**), et qu'une fois cet agent perçu, il le suive partout *même s'il ne se trouve plus dans sa distance de perception*. Écrire le code de la méthode **to move-Suiveur** qui décrit le comportement de **Suiveur**. On supposera qu'un Suiveur avance de manière aléatoire tant qu'il n'a pas trouvé d'agent de type **Leader**. On supposera que **Suiveur** avance à une vitesse de **0.8** et **Leader** a une vitesse de **1.2**.
- b) Supposons qu'il existe un type d'agent appelé **Truc**. Écrire la procédure : **to-report trouverTruc [n]** qui retourne un et un seul agent (ou aucun s'il n'y en a aucun) qui se trouve dans un rayon de **n** patches autour de l'agent courant.

Exercice 3 : Architectures

On veut créer un agent Animal qui a quatre tâches pour décrire son comportement de base :

Tâches de Animal :

1. Éviter un prédateur (lorsqu'il perçoit ce prédateur)
2. Aller à la chasse pour trouver de la nourriture (quand il n'a plus de nourriture pour lui et ses petits)
3. S'occuper de ses petits (quand il a des petits et qu'ils ont faim)
4. Rencontrer un autre Animal pour s'accoupler (il n'y a pas de genre différent chez ces animaux)

Ces tâches sont à effectuer dans cette priorité là (la tâche 1 est la plus prioritaire).

- a) Dessinez le comportement de Animal sous la forme d'un diagramme de Subsumption.
- b) Implémentez Animal en NetLogo sous la forme d'une architecture de subsumption (on ne demande pas le détail des tâches, juste l'architecture générale c'est-à-dire l'organisation générale des procédures à partir de la procédure **to go**).
- c) Dessinez le comportement de Animal sous la forme d'un diagramme FMS (vous pouvez utiliser une variante de FSM comme on l'a vu dans le cours et les TP. Précisez si vous utilisez une telle variante).
- d) Implémenter Animal en NetLogo sous la forme d'une FSM, éventuellement avec sa variante.
- e) Comparez les deux approches, et expliquer l'intérêt de chaque architecture par rapport à l'autre.