

Session :2

Date : 10 / 06 / 2020

Licence ☐ Master M1 Informatique

Mention : INFORMATIQUE

Parcours : IMAGINA, DCOL, AIGLE

Libellé + Code de l'UE : **HMIN 108 – Programmation par agents** Responsable : J. Ferber

Durée de l'épreuve : 2 heures

Documents autorisés : tous

Matériels autorisés : tous

EN LIGNE

SUJET

L'examen portera uniquement sur de la programmation, comme s'il s'agissait d'un TP noté.

- Vous rendrez les fichiers Netlogo nécessaires à la fin en me les envoyant par email sur mon compte ferber@lirmm.fr à la fin de l'examen, dans l'état dans lequel ils sont. Le temps imparti est important.
- Je testerais directement le fonctionnement de vos fichiers.
- Mettez votre nom sur chacun des fichiers netlogo dans le fichier et surtout dans le nom de vos fichier (ex : si vous vous appelez Dupond, vous me rendez des fichiers marqués : dupond-leopard-solo-subsumption et dupond-leopard-solo-FSM pour la 1^{ère} partie et dupond-leopard-accouplement-subsumption et dupond-leopard-accouplement-leopard-FSM pour la 2^{ème} partie).
- Vous pouvez mettre vos remarques sur votre programme dans la partie description des fichiers Netlogo ou dans un fichier word ou text associé.
- Je suis en permanence avec vous et vous pouvez me contacter par le tchat de l'application.

Enoncé : On veut créer un modèle de léopard en NetLogo (un léopard très stylisé en fait!).

1^{ère} partie : solo

On supposera qu'un **Leopard** a trois comportements. Soit il chasse et mange quand il a faim, soit il se désaltère à une source, soit il cherche un abri et va se reposer. Cela définit donc trois tâches: **chasser, boire, reposer**. Le choix de ces comportements est défini par trois propriétés internes au léopard correspondant à un besoin: **faim, soif, sommeil**.

On suppose qu'à chaque boucle, ces valeurs sont incrémentées d'une unité (il a de plus en plus faim, soif et sommeil), et qu'elles redeviennent égales à 0 chaque fois que le léopard respectivement mange, boit et dort (en d'autres termes, s'il mange alors il n'a plus faim. Mais il a encore éventuellement soif et sommeil).

Le léopard choisit la tâche associée à la valeur de la propriété la plus importante si celle-ci est supérieure à une valeur **val-trigger** (la même pour toutes les tâches). Si aucune propriété n'est supérieure à **val-trigger**, le léopard se balade au hasard.

La chasse consiste à suivre une gazelle (on suppose qu'il existe des gazelles dans le système, qui avancent de manière aléatoire) lorsqu'elle apparaît dans le rayon de perception du leopard (**perception-radius**). Dès qu'on est sur le même patch que la gazelle, on suppose qu'il la mange (elle disparaît) et qu'il n'a plus faim.

Boire consiste à chercher au hasard une source. On supposera que les sources sont représentées sous la forme de patches de couleur bleue (**blue**). Dès qu'il est sur une case bleue, il boit et sa soif est mise à 0.

Dormir consiste à chercher une zone marron (brown) et à y rester un nombre de tour égal à **sleep-time** et à remettre à zéro ensuite le sommeil.

On supposera que lorsque un léopard est dans une tâche, il ne prend plus en compte les autres besoins.

Note : Les variables : **val-trigger**, et **perception-radius** sont données dans des sliders.

- a) Implémentez le comportement du Léopard (avec son initialisation) en NetLogo en implémentant directement la machine à état fini en NetLogo.
- b) Implémentez le comportement du léopard à l'aide d'une architecture de subsomption.
- c) Préciser, dans des commentaires ou dans la partie description du fichier NetLog, quelle est la différence entre ces deux approches, quels sont les avantages et inconvénients de ces deux approches dans cet exemple ?

Note: la qualité d'élégance du code sera prise en compte. Relisez bien l'énoncé, il me semble que toutes les informations dont vous pouvez avoir besoin sont données dans l'énoncé.

2^{ème} partie : accomplissement

On suppose maintenant que le Léopard peut s'accoupler. Pour cela, on veut définir le comportement de la femelle et celui du mâle. La femelle émet des odeurs à certaines périodes de l'année qui attire les mâles (cela s'appelle l'oestrus). On supposera que cela se fera tous les **intervalle-reprod** (en nombre de ticks) et l'oestrus dure **temps-oestrus** (en nombre de ticks aussi). A la fin du **temps-oestrus**, le paramètre **intervalle-reprod** est ré-initialisé. On suppose que ces deux paramètres sont placés dans des attributs personnels de la femelle.

Lorsque la femelle est en oestrus, elle émet une odeur, que l'on représentera sous la forme d'une valeur de l'attribut '**odeur-attraction**'. Cette odeur sera diffusée et s'évaporerait avec les taux **diffusion-odeur** et **evaporation-odeur**. Lorsque la femelle est en oestrus, elle arrête ses autres tâches et se déplace aléatoirement. Elle reprend ses autres tâches dès que l'oestrus s'arrête.

Lorsqu'un mâle perçoit cette odeur, il passe en **recherche-accouplement** : il arrête tous les autres comportements et il se dirigera vers la source de l'odeur. S'il ne sent plus d'odeur, il quitte ce mode de **recherche-accouplement**.

Si il rencontre une femelle, alors ils entrent tous les deux en '**accouplement**' (qui dure 1 tick) et cela sort le léopard de **recherche- accouplement**. Il devient ensuite insensible à toute odeur pendant un temps **periode-insensibilité**, ce qui lui permet de reprendre ses autres tâches.

Idem : toutes les variables globales seront représentées sous la forme de sliders.

On pensera en termes de FSM hiérarchique

- d) Donnez le code générique de la FSM hiérarchique du comportement du léopard mâle et femelle.
- e) Donnez le code de l'implémentation des deux types de leopard (mâle et femelle) en NetLogo sous forme de FSM hiérarchique
- f) Donnez le code de l'implémentation des deux types de leopard, à l'aide d'une architecture de subsomption.