Master Informatique 1ère année:

Algorithmes de mouvement

Responsable du module : Jacques Ferber

HMIN233B - Année 2018-2019

Sujet de TD/TP1

TP 1 - Flocking et évitement de collision

On va chercher à développer et comprendre comment on peut créer des algorithmes de formation de groupe et d'évitement de collision. Le problème est celui d'avancer en formation, sans qu'il y ait un chef a priori. On rencontre dans la nature de nombreux exemples de cas où ces formations sont mises en place: trouveaux, bancs de poissons, vols d'oiseaux, etc...

Pour cela on utilisera l'outil NetLogo qui permet de facilement créer des simulations multi-agents. NetLogo a été vu dans le module programmation orientée agent (HMIN108) au premier semestre du M1.

1. Flocking

Aller chercher le modèle Flocking dans la bibliothèque de NetLogo. Tester ses paramètres...

a) Flocking avec des types d'agents différents

On considère maintenant qu'il existe des agents Vert et des agents Bleu. Les Verts ne peuvent créer des groupes qu'avec des Vert et les Bleus avec des Bleus. Voir comment des groupes différents se forment.

b) Flocking qui s'arrête et reprend

On suppose maintenant qu'il existe une troisième espèce d'agent, des agents orange qui eux se déplacnet individuellmenet. On veut que les agents Vert arrêtent de se mettre en formation lorsqu'ils perçoivent, à une certaine distance, un agent orange.

c) Flocking vectoriel

Implémenter l'algorithme de flocking vu en cours avec les trois forces: alignement, écartement, centrage mais maintenant en utilisant la formulation vectorielle...

Vous trouverez les fonctions vectorielles nécessaires dans le fichier fonctions de gestion de vecteurs.

d) Flocking en V

On va essayer maintenant de créer un système de vol en V (comme pour les oiseaux migrateurs)

Pour cela on va appliquer les règles suivantes:

- Règle 1: si un agent est trop loin des autres agents, il accélère pour se rapprocher du plus proche.
- Règle 2: si un agent est suffisamment près d'un autre, il va venir sur l'un de ses côtés, pour que sa vue ne soit pas obstruée
- Règle 3: Si un agent est trop proche d'un autre, il ralentit
- Règle 4: quand les trois autres conditions sont remplies, l'agent adapte sa vitesse et direction à ses voisins visibles

Implémentez l'algorithme permettant le système en V (on pourra partir de l'algorithme de base du flocking).

Note: On rappelle que la fonction in-cone de Netlogo retourne l'ensemble des toriues ou patches qui se trouvent dans un cone de perception. Très pratique pour savoir si un agent se trouve dans un certain cone de perception.