

# **Algorithmes d'exploration et de mouvement**

Intervenant (mais non responsable du module): [Jacques Ferber](#)

**HMIN233B - Année 2018-2019**

## **TP5 - Intégrer algorithmes de recherche et exploration**

On cherche à implémenter un système, dans lequel on a un ensemble d'agents qui se déplacent en utilisant à la fois un algorithme d'exploration et un algorithme de flocking

### **1. Suivre le chemin tracé**

Faire en sorte que l'agent qui a calculé un chemin suive le chemin tracé. Attention, il faudra numéroté les cases du but vers la source pour bien être sûr que l'agent suive le chemin.

### **2. Etre plusieurs à suivre un même chemin**

Implémenter l'algorithme d'exploration pour qu'il serve pour plusieurs agents...

Si un agent perçoit un chemin, alors il le suit..

Sinon il commence à calculer son algorithme d'exploration (A\* ou Dijkstra. S'il rencontre un chemin, alors il ne va pas jusqu'au but, mais il considère que le chemin trouvé est le but, et il "remonte" vers la source...

Implémentez cet algorithme et faites en sorte que les agents avancent vers le but à partir de cet algorithme

### **3. Calculer un chemin "dans la tête"**

(sauvegarder le fichier de l'algorithme d'exploration précédent, car la structure d'exécution est très modifiée)

Maintenant, on modifie l'algorithme pour que le chemin calculé devienne un plan de l'agent (et non plus un tracé de l'environnement). Pour ce faire, on crée un attribut 'plan' dans l'agent et les chemin calculés par draw-path sont retournés sous la forme d'une liste qui sera placé dans cet attribut.

Modifier l'algorithme de déplacement de la question 1) pour que l'agent suive son plan et aille au but en suivant son plan.

Faire en sorte que l'agent n'avance que d'une étape du plan (une case) à chaque tick ! Attention: vous aurez besoin d'un autre attribut pour stocker le plan en cours d'exécution.

## 4. Avancer en suivant le chemin de l'autre

On va implémenter un système dans lequel les agents peuvent utiliser le plan des autres

Execution avec voisins:

soit  $E$  l'ensemble des voisins,

si j'ai un plan, j'exécute ce plan, stop

si j'ai un voisin  $x$  de  $E$  qui a un plan, alors j'exécute le plan de  $x$ , stop

sinon je calcule un plan, je signale que j'ai un plan,  
et j'exécute ce plan, stop

Implémentez ce système... On verra ensuite quelques variantes pour l'améliorer

## 5. Avancer en suivant le chemin de l'autre en formation

Intégrer le flocking au système précédent