Master Informatique 1ère année:

Algorithmes d'exploration et de mouvement

Intervenant (mais non responsable du module): Jacques Ferber

HMIN233B - Année 2018-2019

TP2 - Evitement de collision

1. Evitement de collisions statiques

On désire que ce groupe d'agents puisse éviter des obstacles. Pour réaliser cela, on utilisera le fichier NetLogo de template où il suffit d'implémenter la procedure avoid-obstacles

a) Fuite

En partant de la base du flocking standard (choisir d'abord le modèle standard et ensuite le modèle vectoriel), implémentez un algorithme d'évitement standard qui consiste à aller dans la direction opposée quand un agent rencontre un obstacle. On parle alors de fuite.

b) Evitement proprement dit

Quand un obstacle entre dans son rayon de perception l'agent cherche à éviter l'obstacle. Pour cela l'agent se détourne de sa direction initiale d'un angle inférieur à un angle max (max-avoidance-turn). On verra que dans ce cas, il est possible que les agents entrent en collision avec les obstacles. Dans ce cas, implémentez, lorsque la distance est vraiment trop courte (environ 2 à 3 unités), un comportement de fuite.

Notez les différences entre modèle standard (algo) et vectoriel

2. Créer un champ de potentiel fluide

On suppose qu'il existe un ensemble d'obstacles fixes et de buts fixes. On va essayer de créer dynamiquement un champ de potentiel considéré comme une vague.

Chaque but propage des valeurs qui vont se diffuser dans l'environnement et simplement contourner les obstacles statiques.

Adaptez l'algorithme d'inondation de manière à prendre en compte un poteniel négatif des obstacles.

Tester votre algorithme en faisant circuler un ensemble d'agents sur ces champs de potentiel.

Principe de l'algorithme:

```
While continue [
    set continue false
    ask patches [inonder]
]
```

Ecrivez le code de l'algorithme d'inondation à partir du template proposé ici.

a) Question: écrire le code de la fonction inonder qui crée un champ de potentiel qui contourne les obstacles

b) Adapter cette fonction de manière à faire en sorte que les obstacles soient des repoussoir de qulques pixels, en utilisant le même principe, mais à partir des obstacles qui diffusent un potentiel négatif.

3. Evitement de collisions dynamiques

On considère maintenant que chaque agent est un cercle définit par une taille fixe (le rayon du cercle). On resuppose qu'il existe des agents Verts et des agents Bleus de manière à avoir des groupes de couleur différente.

Modifiez les algorithmes d'évitement d'obstacles pour qu'ils prennent en compte l'évitement des agents les uns vis à vis des autres, et notamment que les Bleus évitent les Vert et réciproquement.

a) Evitement d'agents

Implémentez, avec l'algorithme "standard" l'évitement par répulsion entre les agents bleus et verts.

b) Evitement d'agents

partir d'une souche 2.b ou 2.c du TP1 (évitement proprement dit, avec priorité ou contrôle par vecteur), une technique anti-collision entre les agents.

b) Algorithme d'évitement fondé sur la notion de "forces de glissement"

Implémentez l'algorithme fonctionnant à partir des forces de glissement (on utilisera la technique vectorielle vue auparavant). Ici le vecteur **E** est défini comme étant perpendiculaire au vecteur qui relie l'agent à l'obstacle (heading ± 90). Le coté du glissement est donné par la différence vectorielle (voir cours).