Projet TER M1

EvoAgents

Soutenu par :

Les Québécois

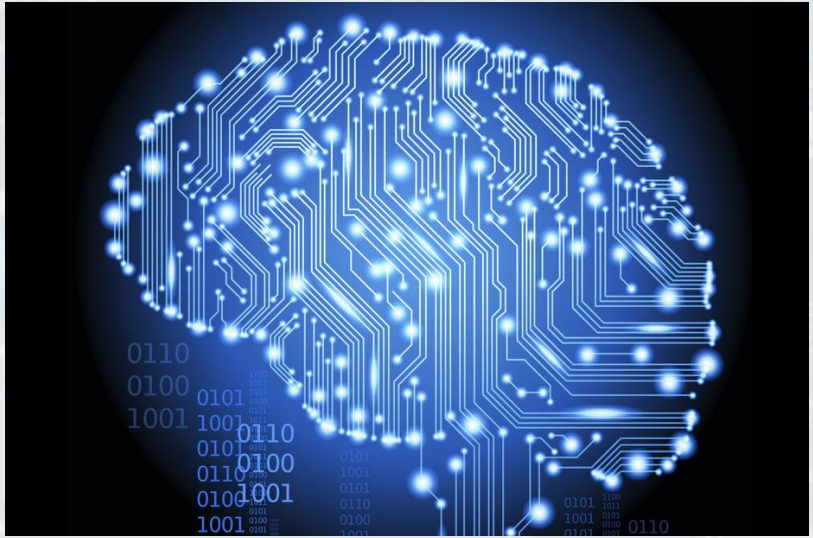
Bourgeois Homère, Courbier Raphaël,
Gautier Corentin, Odorico Thibault

Encadré par :

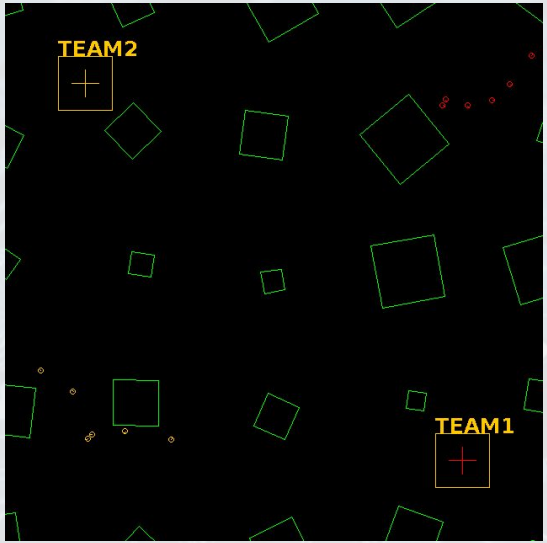
Suro François
Ferber Jacques

2018-2019

Sommaire



Introduction



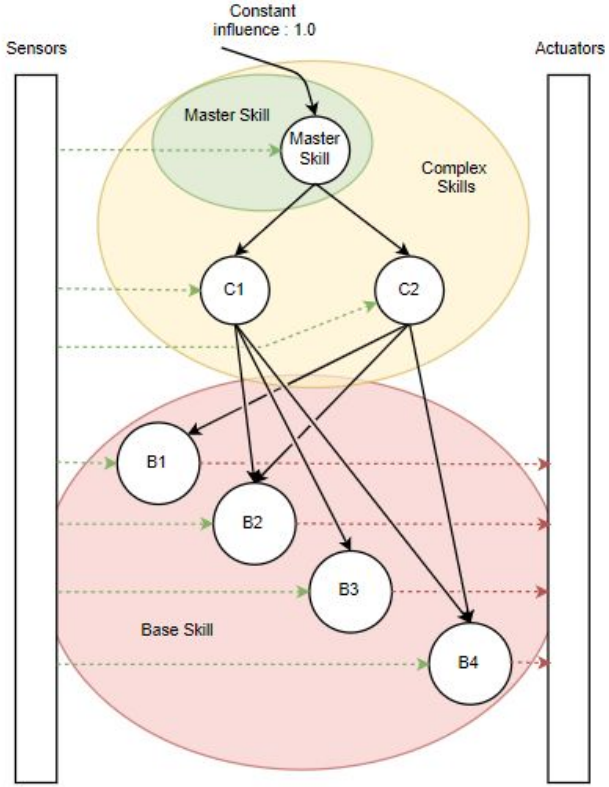
Environnement de capture de drapeaux

- Réseau de neurones
- Hiérarchie MIND
- Présentation du projet

Problématique : L'utilisation de la hiérarchie MIND est-elle toujours pertinente dans le cadre d'un système multi-agents ?

Hiérarchie MIND

- Plusieurs types de compétences
  - Base skill
  - Complex skill
  - Master skill
- La complex skill influence ses sub-skills
- Entrainement avec apprentissage génétique



Hiérarchie de capacitées générique

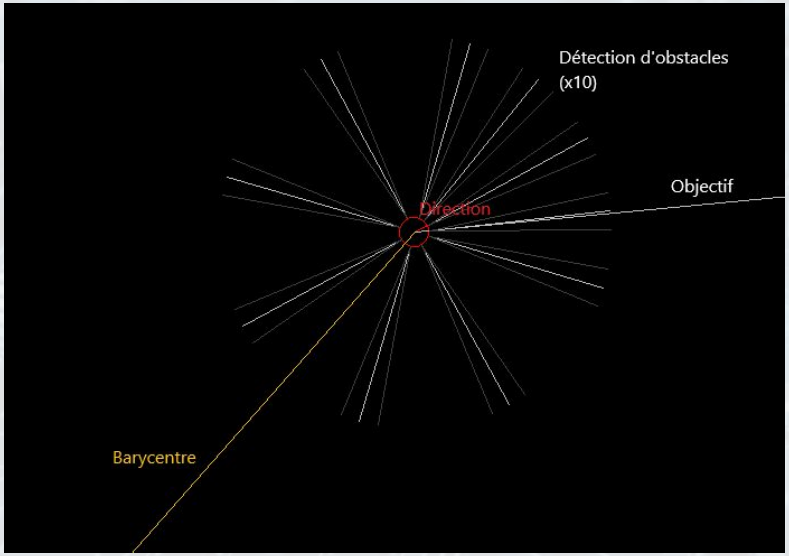
Apprentissage EvoAgents

```
makeBot();

target = new Waypoint(new Vec2(), 0, TARGET_SIZE);
getWorldElements().add(target);

((S_Radar)bot.sensors.get("TargetOrient")).setTarget(target);
((S_Distance)bot.sensors.get("TargetDistance")).setTarget(target);

controlFunctions.add(new CF_NextOnMovingFarFromStartPosition(bot, this, 4));

rewardFunctions.add(new RW_SameOrientationAsTarget(bot, target, 20));
rewardFunctions.add(new RW_StayOnPlace(bot, 1));

initTraining(training);

makeWorld();
```

Code d'entraînement pour l'orientation

- **Environnement** (paramètres de lancement)
- **Fonctions de récompenses**
  - Gagne ou perd des points
- **Fonctions de contrôles**
  - Paramètres
  - Fin d'un apprentissage
- **Cluster**
  - Multithreading
  - Serveur

Entraînement des agents

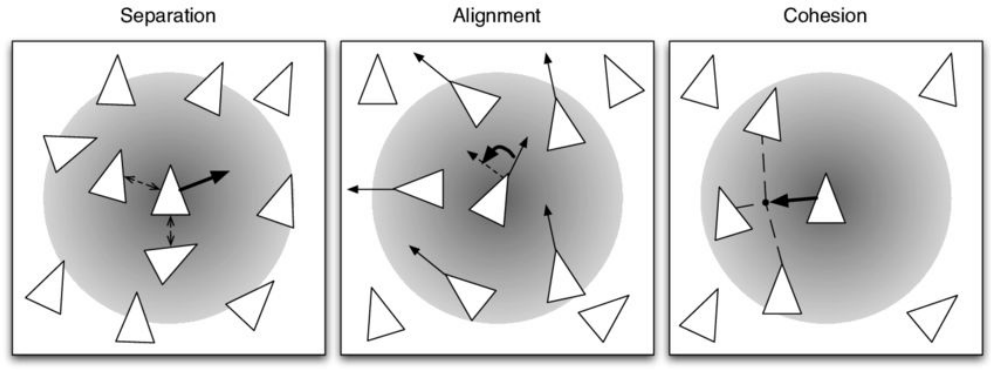


Le robot d'entraînement

- **Sensors**
  - Détection d’obstacles
  - Détection d'ennemis/alliés
  - Détection de cible (Type, distance, orientation)
- **Actuators**
  - Moteur des roues
  - Gestion du canon (orientation, rechargement, tir)

Stratégies de développement

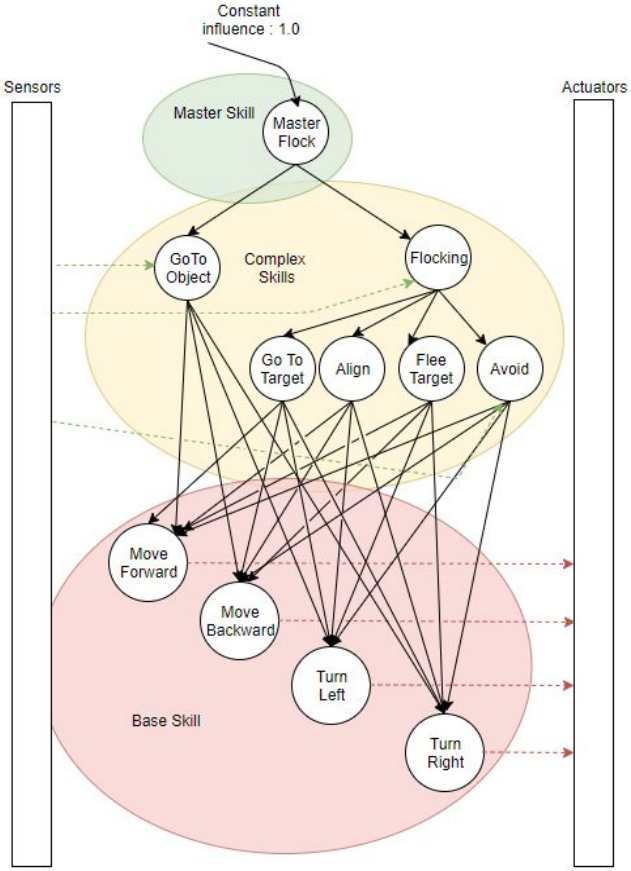
- Changement de la stratégie finale
- Mouvement de Flocking



Composantes du flocking

Stratégies de développement

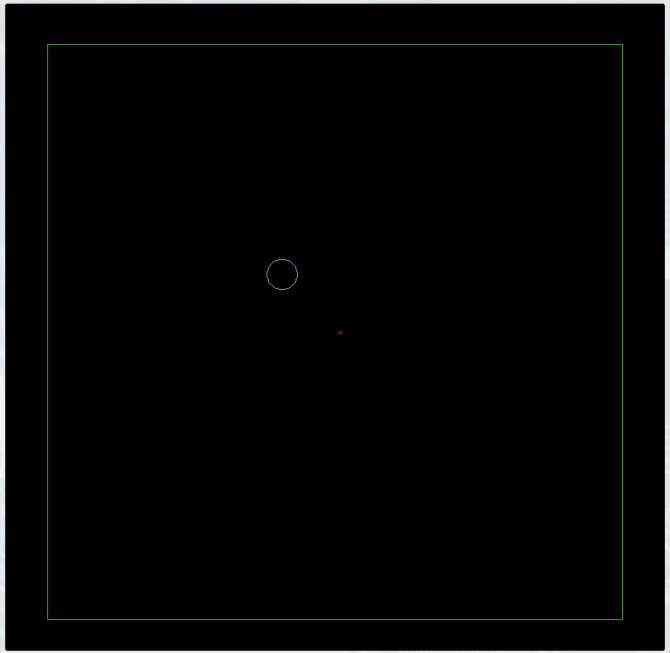
- **GoToObject** : Se dirige vers un objectif
- **Avoid** : Évite les obstacles
- **GoToTarget** : Se dirige vers la barycentre
- **Flee** : Se sépare du barycentre
- **Align** : Adapte sa vitesse et son orientation
- **Flock** : Mouvement de groupe



Hiérarchie finale

Environnement : Flee

- **Environnement** :
  - Monde sans obstacles de grande taille
  - Cible à fuir
- **Récompenses** :
  - La distance avec la cible doit augmenter
  - La vitesse du bot doit être maximale
  - Le bot doit se déplacer en avant
- **Exercice** :
  - Cible se génère à une position aléatoire
  - Répétition de l'exercice s'il est réussi



Environnement d'entraînement “Flee”

Imprévus : Flee

```
FleeTarget
input:4
VA:VAR_TARGET_FRIENDLY_ORIENT
VA:VAR_TARGET_FRIENDLY_DISTANCE
VD:VAR_TARGET_FRIENDLY_ORIENT
VD:VAR_TARGET_FRIENDLY_DISTANCE
output:4
SK:moveForward
SK:turnLeft
SK:turnRight
SK:moveBackwards
neural
layers:2
```

Fichier de description de “Flee”

- Erreur de définition des variables
- Recalibrage des valeurs des fonctions de récompenses

Démonstration : Flee

- Comportement satisfaisant
- Possibilités d'améliorations du demi-tour

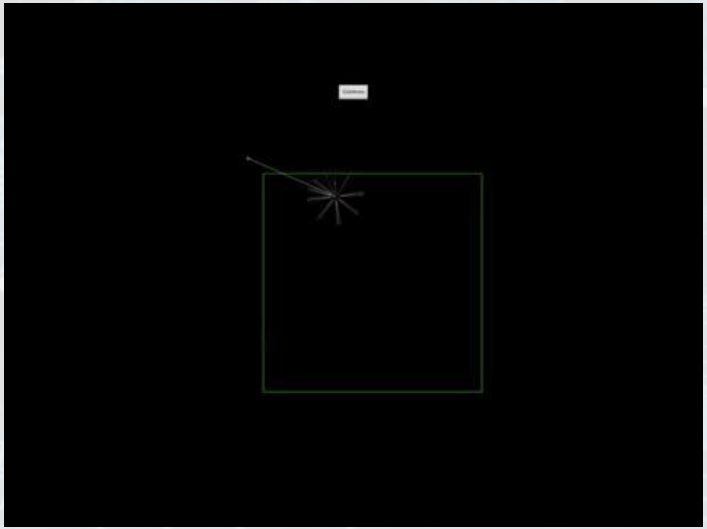
Vidéo : Flee

Environnement : Align

- **Environnement** :
  - Environnement single-agent
  - Monde vide de taille infinis
  - Cible avec laquelle s'aligner
- **Récompenses** :
  - La distance avec la cible doit rester la même
  - L'orientation du bot et de la cible doit rester la même
  - La vitesse du bot et de la cible doit rester la même
  - Le bot doit se déplacer en avant
- **Exercices** :
  - Une cible aléatoire avance dans une direction aléatoire à une vitesse aléatoire

Imprévus : Align

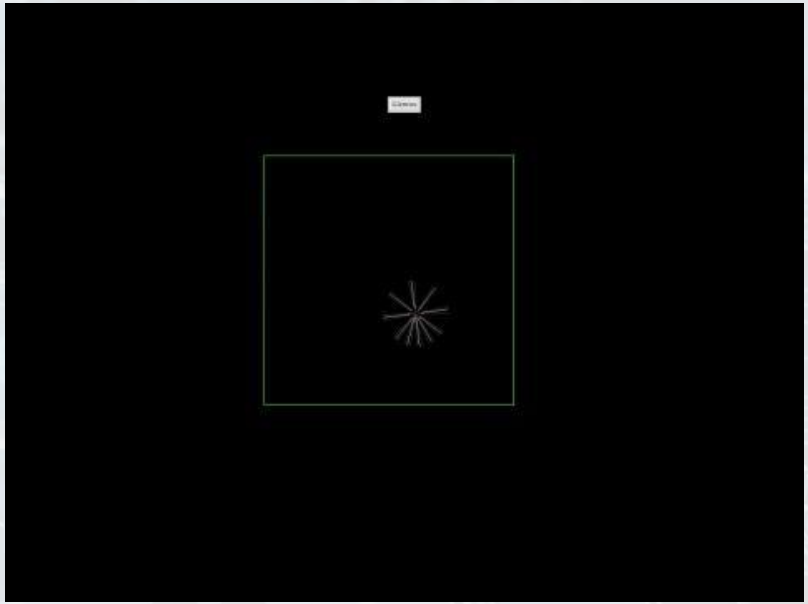
- **Découverte du robot :** Suivre la cible, est un comportement simple qui finit par payer !



Vidéo : Align “suiveur”

Modifications : Align

- **Création d'exercices :**
  - Cible qui se déplace vers le robot
  - Cible qui orbite autour du robot
  - Cible qui reste fixe



Vidéo : Exercices d'alignement

Modifications : Align

- **Création de sub-skills :**

Vidéo : KeepSameSpeedAsTarget

Vidéo : KeepSameOrientationAsTarget

Démonstration : Align

- La modularité de MIND permet de décomposer une capacité facilement et de cibler les difficultées du robot

Vidéo : Align version finale

Environnement : Flock

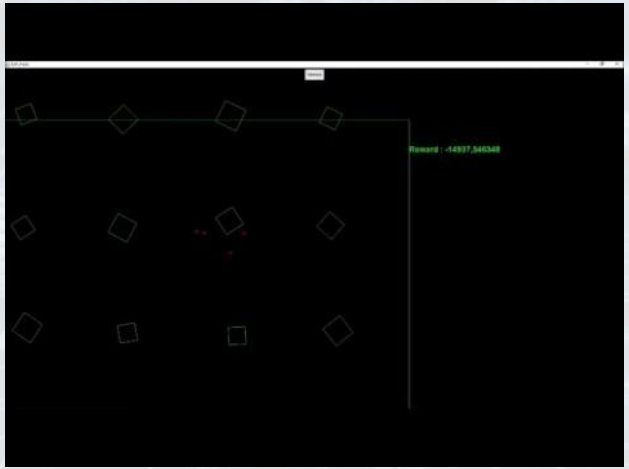
- **Environnement :**
  - Environnement multi-agent
  - Environnement vide.
  - Cible immobile à atteindre.
- **Récompenses :**
  - Rester proche du centre du groupe.
  - Rester assez éloigné des autres agents du groupe.
  - Aller vers la cible.
  - 2ème expérience : Punition si collision.
- **Exercice :**
  - Aller en groupe vers une cible fixe.

Imprévus : Flock

Vidéo : Flocking sans obstacles

- Premiers résultats encourageants !
- Cependant, problème d'analyse : le comportement voulu est-il bien appris?

Imprévus : Flock



Vidéo : Flocking avec obstacles

- Test avec des obstacles
- Problème : cohésion entre les robots
- Conflit entre évitement d'obstacle et flocking

Conclusion

- Succès de la hiérarchie MIND dans un système multi-agents ?
  - Généricité
  - Encapsulation
  - Flexibilité
- Difficultées / Améliorations ?
  - Moins de temps de conception
  - Différentes stratégies d'apprentissage

Merci pour votre attention