



Université Claude Bernard



Lyon 1

Master 2 STS mention Informatique

Parcours «Intelligence Artificielle»

---

# Rapport de TP

## TP 2 SMA : Robots de tri

### Version 1

---

*Auteurs :*

Mourtaza ABBASBHAY  
Brice MARC

*Encadrants :*

M<sup>me</sup> Salima HASSAS

Version du  
7 novembre 2021

## **Table des matières**

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>L'environnement</b>	<b>2</b>
<b>3</b>	<b>L'agent</b>	<b>3</b>
<b>4</b>	<b>La simulation</b>	<b>3</b>
<b>5</b>	<b>Influence des paramètres</b>	<b>4</b>
<b>6</b>	<b>Résultats</b>	<b>4</b>
<b>7</b>	<b>Conclusion</b>	<b>5</b>

## 1 Introduction

Dans ce TP, on souhaite programmer des agents pour qu'ils réalisent le tri collectif d'objets répartis dans leur environnement. On veut se limiter à des comportements basiques de type perception/action, les agents sont purement réactifs. Ils ne disposent que d'une quantité minimale d'informations et ne communiquent pas entre eux. Ils n'interagissent qu'avec leur environnement proche et ont une mémoire limitée.

## 2 L'environnement

L'environnement est représenté par une grille contenant les agents et les objets. Il y a deux types d'objets notés A et B par la suite. Ces éléments sont initialement positionnés aléatoirement sur la grille. En tout instant, deux objets ne peuvent pas se trouver sur la même case, de même pour deux agents. Un agent peut cependant être sur la même case qu'un objet.

Les agents n'ont pas connaissance de leur position dans l'environnement car ils sont totalement réactifs. Leur position est donc mémorisée par l'environnement. Lorsque l'agent effectue une perception, l'environnement ne l'informe pas de sa position ni de la présence d'autres agents, il lui indique uniquement la présence ou non d'un objet sur sa case, et le type A ou B de l'objet s'il y en a un.

Lorsqu'un agent prend un objet, l'environnement supprime l'objet de la grille et confie sa propriété à l'agent. Puis lorsque l'agent veut déposer l'objet, l'environnement l'autorise ou non à le déposer s'il n'y a pas déjà un objet sur la case. On pourrait confier la décision à l'agent car il sait s'il y a un objet à son emplacement grâce à la perception qu'il effectue. Cependant comme il est uniquement réactif, on le laisse plutôt tenter de déposer l'objet, et c'est ensuite l'environnement qui l'en empêche. Si l'emplacement est libre, l'agent restitue la propriété de l'objet à l'environnement.

De même lorsque l'agent tente de se déplacer, l'environnement l'autorise à bouger s'il reste dans la grille et s'il n'y a pas d'autre agent à sa destination. L'agent n'a pas connaissance de ces informations, c'est l'environnement qui l'autorise ou non à se déplacer.

### 3 L'agent

L'agent peut percevoir son environnement proche mais pas les autres agents, et il ne connaît pas sa position globale. L'agent possède une mémoire à court terme dans laquelle il retient le contenu des dernières cases qu'il a visité (objet A ou B ou rien). Cette mémoire modifie son comportement en présence d'un objet. L'agent sait s'il transporte un objet et de quel type d'objet il s'agit. Son comportement est toujours réactif, il réagit aux dernières cases qu'il a visité et à l'objet qu'il transporte éventuellement.

L'agent effectue une action selon des seuils aléatoires qui dépendent de son état et de sa mémoire. Il a plus de chances de prendre un objet si les dernières cases qu'il a visité ne présentaient pas d'objets du même type. Inversement, il a plus de chances de déposer un objet si les dernières cases visitées présentaient un objet du même type. Ainsi il prend les objets isolés et les dépose dans des zones plus adaptées.

L'agent se déplace selon un pas constant et dans une direction aléatoire. Si la direction choisie est incompatible avec l'environnement, l'agent reste sur place et attend d'effectuer sa prochaine action.

### 4 La simulation

La simulation évolue au tour par tour. A chaque tour, les agents agissent dans un ordre aléatoire. Chaque agent commence par effectuer une perception, l'environnement l'informe de ce qui se trouve sur sa case. Il effectue ensuite une action parmi la prise d'un objet, le dépôt d'un objet, ou le déplacement aléatoire. Un tirage aléatoire a lieu à chaque tour selon les seuils pour savoir s'il prend ou dépose un objet. S'il ne fait rien, il se déplace.

Une fonction score peut être utilisée sur la grille d'objets pour évaluer à quel point la grille est triée. Pour chaque objet, on incrémente le score de 1 pour chaque objet du même type se trouvant sur une des 8 cases adjacentes. On estime ainsi si à quel point les paquets sont bien regroupés en paquets du même type, ce qui est l'effet recherché par le tri des agents.

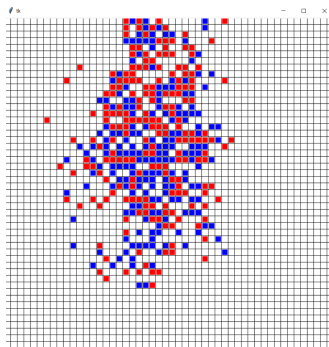
Une interface Tkinter est utilisée pour visualiser l'évolution de la position des objets pendant la simulation (intervalle de mise à jour modifiable).

## 5 Influence des paramètres

La taille de la mémoire joue sur l'étendue et la précision du tri réalisé par l'agent. Plus la mémoire est large, plus l'agent va trier une zone large mais il perd en précision, c'est à dire que les objets du même type ne seront pas côte à côte mais répartis sur une zone. Mais comme les agents se déplacent aléatoirement, le concept de zone n'a pas vraiment de sens. Il faudrait affecter à chaque agent une zone hors de laquelle il ne doit pas sortir et dont il est responsable du tri.

Deux paramètres  $k1$  et  $k2$  modifient les probabilités de prise et de dépôt de l'agent au contact d'un objet. On peut ainsi augmenter ou diminuer l'influence de la mémoire sur le comportement de l'agent.

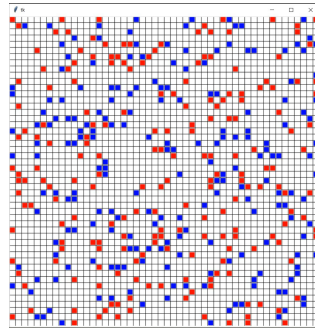
On peut aussi affecter un taux d'erreur lors de l'identification d'un objet par l'agent. Plus ce taux est élevé, moins le tri est performant. Pour un taux d'erreur de 100%, les agents vont regrouper les objets sans pouvoir tenir compte de leur type. On obtient alors après un certain temps un unique paquet, tous les objets sont regroupés ensemble. Un exemple est visible sur la figure 1. Le score après 100 000 itérations est de 738, à comparer avec les résultats qui vont suivre.



**FIGURE 1** – Résultat pour un taux d'erreur de 100%

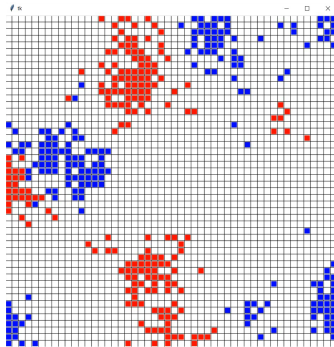
## 6 Résultats

Les simulations sont lancées à partir de la position initiale suivante, la seed aléatoire est la même lors des essais.

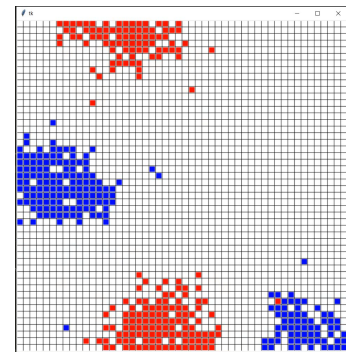


**FIGURE 2** – Initialisation de la simulation

Après avoir laissé tourner la simulation, on obtient les résultats suivants :



**FIGURE 3** – Résultat après 100 000 itérations



**FIGURE 4** – Résultat après 1 000 000 d'itérations

On constate que la simulation converge, les objets sont bien regroupés en paquets du même type. On n'arrive pas à rassembler les 4 derniers paquets en 2 car les paquets de même type sont trop éloignés l'un de l'autre et un paquet de type opposé se trouve au milieu et fait donc obstacle au tri. Le score initial est de 255. Après 100 000 itérations, il est de 1 433 et après 1 000 000 itérations, il atteint 2 066. Le score augmente bien plus les objets sont triés.

## 7 Conclusion

Grâce à cette simulation, on arrive à trier des objets dans l'espace en utilisant des agents uniquement réactifs qui n'ont pas connaissance de leur position, ne peuvent observer que leur environnement proche, et ne communiquent pas entre eux.