



Université Claude Bernard



# **Rapport de TP**

## **TP 2 SMA : Tri Collectif**

### **Version 2**

***Réalisé par :***

*khalid OUHMAID*

*Mohamed Amin ASRI*

***Encadrants :***

Mme Salima HASSAS

2021/2022

# Table des matières

|  |           |
|--|-----------|
| <b>Présentation .....</b>                        | <b>3</b>  |
| <b>Mise en place .....</b>                       | <b>3</b>  |
| <b>Architecture Voyelle.....</b>                 | <b>4</b>  |
| <b>Introduction d'objets C .....</b>             | <b>4</b>  |
| <b>Les objets coopératifs .....</b>              | <b>5</b>  |
| <b>L'agent.....</b>                              | <b>6</b>  |
| <b>La simulation .....</b>                       | <b>7</b>  |
| <b>Influence des paramètres.....</b>             | <b>7</b>  |
| <b>Mécanisme de collaboration – Fusion .....</b> | <b>8</b>  |
| <b>Résultats obtenus .....</b>                   | <b>9</b>  |
| <b>Conclusion .....</b>                          | <b>13</b> |

# Présentation

Le TP consiste à réaliser un algorithme simple de tri-collectif. Cet **algorithme est inspiré du comportement de certaines colonies de fourmis** triant leurs larves. Les agents se déplacent au hasard, ne **communiquent pas**, n'ont pas d'organisation hiérarchique, pas de représentation globale et ils ne peuvent percevoir que des objets juste devant eux. Ils peuvent cependant distinguer les différents types d'objets avec un certain degré d'erreur. La probabilité qu'ils ramassent ou déposent un objet est modulée en fonction du nombre d'objets identiques qu'ils ont rencontrés dans un passé récent. Cela crée un feed-back positif suffisant pour coordonner l'action des robots. Bien que moins efficace qu'un tri hiérarchisé, cette organisation décentralisée offre des avantages de simplicité, de flexibilité et de robustesse.

## Mise en place

Soit un environnement représenté par une grille sur lequel on dispose initialement aléatoirement, des objets identifiés par des lettres A ou B et un certain nombre d'agents. Le déplacement des agents est aléatoire et la prise ou le dépôt d'un objet dans l'environnement sont conditionnés respectivement par une probabilité de prise et une probabilité de dépôt qui s'exprime respectivement par :  $P_{prise} = (k_+ / (k_+ + f))^2$  et  $P_{dépôt} = (f / (k_- + f))^2$  avec :  $k_+$  et  $k_-$  des constantes et  $f$  représentant la proportion d'objet de même type A ou B rencontrés qu'il a gardé dans une mémoire à court terme. Cette mémoire à court terme fonctionne comme une pile FIFO qui stocke les objet rencontrés lors des  $n$  derniers pas. Si l'agent a une mémoire des 10 derniers pas et que lors de ces 10 derniers pas le robot a rencontrés 3 objets A et 2 objets B alors  $f_a = 3/10$  et  $f_b = 2/10$ . Le fonctionnement de l'agent se résume de manière simple par une boucle de perception de son environnement puis d'action, rien de plus. Le tri sera évalué en réalisant regardant la proportion d'objets de même type dans un voisinage direct de chaque objet. Chaque case en dehors des bords et des coins a 8 voisins directs. Si sur la case il y a un objet A et dans son voisinage direct 4 objets A et 2 objets B alors, le score associé à cet objet sera de  $(4-2)/8$ . On calculera la moyenne pour les objets B et pour les objets A. La valeur maximum théorique est très proche de 1 si tous les objets B sont regroupés entre eux, seuls les objets aux bords ayant pour voisins de objets B et du vide ne vont pas avoir un score de 1.

# Architecture Voyelle

## A, Environnement

- **Grille** où se déplace les agents et où sont placés les objets.
- **Objets A, B, C**, ces objets font partis de l'environnement et leurs méthodes sont gérées par l'environnement.

## B, Agents

- **Agents réactifs** : agents qui agissent se déplacent au hasard ou suivent le gradient de phéromone si la trace est assez forte.
- Ils sont munis de fonctions de **perception** (regard au sol) **déplacement**, de **prise et de dépôt d'un objet**. Ces seules fonctionnalités leurs permettent de triés efficacement les tas d'objets ne nécessitant pas d'interaction (A et B). Pour trié les objets C d'autres fonctionnalités leurs seront implémentés (cf partie IV).

## C, Interactions

- Pas d'interaction dans la première partie.
- ## D, Organisation
- **Organisation émergente** : Sans que ces comportements soient codés, les agents vont s'auto-organiser pour triés les objets.

# Introduction d'objets C

On souhaite ajouter un nouveau comportement collectif aux robots. Pour cela on ajoute  $n_C$  objets de type C, qui nécessitent la collaboration de 2 robots pour être portés. Dans ce cas, quand un robot tombe sur un objet C dans son environnement, il émet un phéromone (appel à l'aide) qui se code par une information propagée dans son voisinage (8 cases autour) sur une distance de diffusion **DS** (ex : les 24 cases autour de l'agent si  $ds=2$  dans ce cas pour un agent se trouvant à l'emplacement  $(i,j)$  on aura toutes les cases se trouvant à  $i+k, j+l$ , avec  $0 \leq k \leq 2$ , et  $0 \leq l \leq 2$ ). Il faut noter que l'intensité du phéromone se réduit au fur et à mesure qu'on s'éloigne de la case où se trouve l'agent. Sur la case de l'agent, l'intensité du phéromone est de 1, puis de  $1/2$  pour les cases à distance 1, puis  $1/3$  pour les cases à distance 2 etc, ... ( $1/\text{Norme infini}(\text{pos1-pos2})$ ). La phéromone s'évapore d'un taux **R** à chaque itération. Le robot qui émet des phéromones peut réémettre ou abandonner la tâche au bout d'un certain temps **T**, pour éviter des situations d'interblocage (les robots sont bloqués à attendre de l'aide, alors qu'il pourraient débloquer la situation en s'aidant mutuellement l'un après l'autre). Pour pouvoir atteindre les zones où la quantité de phéromone est la plus importante, il faut munir les agents de capacités sensorielles. On ajoute à la fonction de perception (initialement elle ne permet que de voir les objets immédiatement au sol)

une capacité de percevoir les taux de phéromones dans les 8 cases directement voisines. Lors de son déplacement, si il ne transporte pas d'objets, l'agent va suivre la direction où la quantité de phéromone est le plus fort avec une probabilité  $p=(1-\exp(-\text{pheromon}))$ , cela signifie que si la quantité de phéromone sur un case est faible, cette probabilité sera très proche de 0 alors que si la quantité de phéromone est grande, la probabilité de suivre ce chemin sera très grande. Dans le cas où l'agent ne se dirige pas dans la direction où la quantité de phéromone est la plus importante, il continue de suivre un comportement aléatoire. Lorsqu'un agent arrive sur une case présentant un objet C, il vérifie par la même occasion si un autre agent se trouvant sur la case serait susceptible de l'aider (agent ne transportant pas d'autres objets). Si c'est le cas, il va y avoir **coordination** des 2 agents pour pouvoir transporter l'objet. L'agent qui vient d'arriver sur la case va être désigné comme le **chef de la relation bilatérale** entre les 2 agents et va régir le comportement de l'association. Le couple d'agent se déplacera selon les bons vouloir de l'agent chef et l'association ne prendra fin que lorsque l'agent chef aura décidé de déposer l'objet C. A partir de ce moment là, les 2 agents seront de nouveaux libre de leurs actions. Si l'agent ne trouve pas d'aide sur la case, il appelle à l'aide et attend T itérations jusqu'à ce que quelqu'un susceptible de l'aider arrive. Au bout de T itérations, il s'en va. Une des choses à prendre en compte dans la modélisation présentée jusque là est que si un agent abandonne, il va tout de même être tenté de revenir vers la case où il se trouvait de part la forte quantité de phéromone qui se trouve et ceux malgré le phénomène d'évaporation. Pour prendre en compte ce phénomène, il faut donc faire en sorte que les agents ignorent les informations concernant les phéromones pendant une durée **D** après avoir abandonné la prise d'un objet C.

## Les objets coopératifs

Il y a trois types d'objets présents dans l'environnement. Les objets A et B peuvent être déplacés par un seul agent, mais les objets C nécessitent la collaboration de 2 robots pour être portés. Comme les agents ne communiquent pas entre eux, la présence d'un agent près de l'objet est détecté comme si c'était un nouveau type d'objet. Selon la situation dans laquelle l'objet C se trouve, il est perçu différemment par les robots.

Les objets C peuvent se trouver dans 5 états :

- C0 : aucun robot ne transporte l'objet
- C1 : un robot a pris l'objet et attend de l'aide
- C2 : un second robot est arrivé pour aider, mais le premier robot ne l'a pas encore détecté
- C3 : le robot précédemment en C1 a détecté le second robot arrivé en C2 et il peuvent donc déplacer l'objet
- C4 : un des deux robots a lâché l'objet mais le second ne l'a pas encore détecté

Ainsi l'objet est retiré de la grille lorsque l'étape C3 a lieu. Les deux robots sont alors en possession de l'objet et se déplacent ensemble. Pour différencier les robots, l'un transporte l'objet C2 et l'autre C3, mais l'inverse pourrait aussi être possible, cela représente seulement l'ordre dans lequel ils l'ont pris. Cela permet d'éviter qu'un robot n'essaie de bouger avant que son partenaire ne l'ait détecté. Lorsque l'un des deux robots le lâche et qu'il passe en C4, l'objet

est retourné à l'environnement.

Les causes de changement d'état perçu d'un objet C sont les suivantes :

- C0 -> C1 : un robot prend l'objet
- C1 -> C0 : l'attente a été trop longue et le robot dépose l'objet
- C1 -> C2 : un deuxième robot prend l'objet
- C2 -> C3 : le premier robot détecte l'arrivée du deuxième et confirme
- C2 -> C4 : le second robot lâche l'objet
- C3 -> C4 : le premier robot lâche l'objet
- C4 -> C0 : le robot restant détecte que l'autre l'a lâché et lâche aussi l'objet

Dans l'environnement, on ne voit donc que des objets C0 et C1, et parfois pendant une itération un objet C2 ou C4 le temps que le robot détecte le changement. Les objets C2 et C3 sont transportés par les robots. Lorsque le second robot arrive pour aider le premier et que l'objet passe de l'état C1 à C2, un couple constitué des identifiants des deux robots est enregistré dans l'environnement. Par la suite et tant qu'aucun des deux ne lâche l'objet, ils bougent ensemble. Quand un robot se déplace, il entraîne avec lui le second robot. Il n'y a pas de meneur, les robots agissent normalement chacun leur tour. Pour conserver la continuité de la mémoire, on enregistre aussi la perception des objets proches dans la mémoire du robot qui est entraîné par l'autre. La présence d'un objet C1 et d'un seul robot sur une case est l'unique raison pouvant autoriser les robots à s'y déplacer, sinon il ne peut pas y avoir deux agents sur la même case (sauf après C4).

## 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, B, C 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. 4.1 Ajout de la V2 5 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.

Si l'agent choisit de prendre un objet C0, il passe en mode attente. Dans ce cas, il va passer les prochains tours à attendre de l'aide. Il ne remplit pas sa mémoire inutilement avec le même objet si il attend. Pendant l'attente, le robot émet un signal sonore à un volume fixe sur sa case qui se propage avec atténuation ensuite. Il dispose d'un temps d'attente maximal au bout duquel il va lâcher l'objet. Si l'objet perçu est différent pendant l'attente, il est enregistré en mémoire ce qui va changer son action, car un robot est venu l'aider. Il va alors mettre fin à l'attente et détecter cette aide, l'objet passe alors à C3 et il le prend. Si l'agent était en transport mais qu'il percevait un objet

C4 sur sa case, c'est que son partenaire a lâché l'objet. Il change donc lui aussi son objet transporté en C4 pour modifier son action qui va alors être le dépôt de l'objet. Quand un agent transporte un objet C, sa probabilité de le lâcher est divisée par 2 car ils

sont 2 porteurs pour éviter trop d'abandons. L'agent perçoit le son sur sa case et les 8 cases adjacentes. Si il ne transporte rien, il se déplace dans la direction où le son est le plus élevé et si ce son est plus élevé que sa position actuelle. Sinon, l'agent se déplace dans une direction aléatoire d'un pas constant. Si la direction choisie est incompatible avec l'environnement, l'agent reste sur place et attend d'effectuer sa prochaine action.<sup>3</sup>

## La simulation

La simulation évolue au tour par tour. Les agents agissent chacun leur tour dans le même ordre. Chaque agent commence par effectuer une perception, l'environnement l'informe de ce qui se trouve sur sa case et du son autour de lui. Il effectue ensuite une action parmi la prise d'un objet, le dépôt d'un objet, l'attente, le déplacement vers le son 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). Les objets A sont rouges, les B bleus et les C verts.

## Influence des paramètres

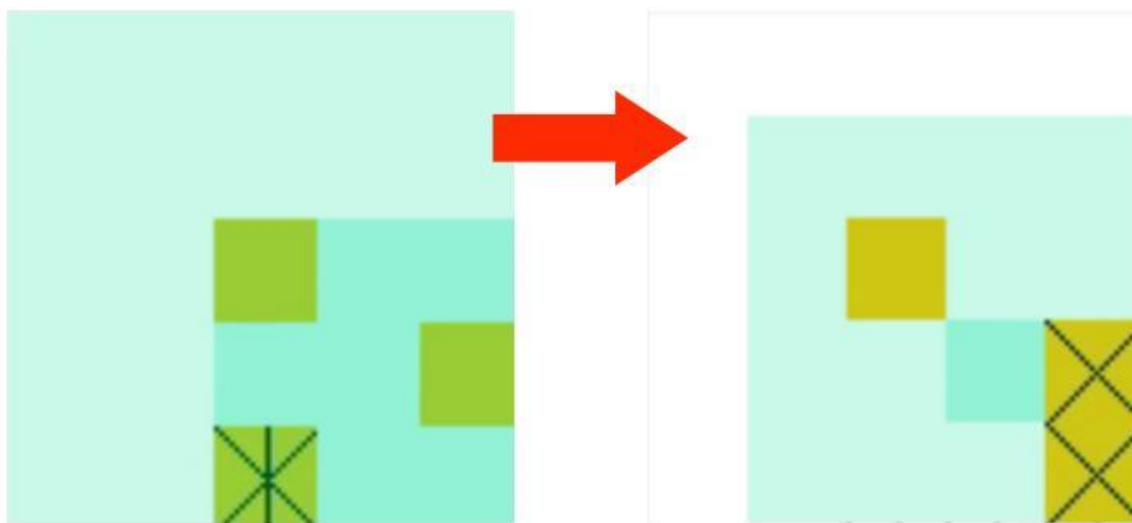
La taille de la mémoire joue sur l'étendue et la précision du tri réalisé par l'agent. Plus la mémoire est large, plus l'agent va trier une zone large mais il perd en précision, c'est à dire que les objets du même type ne seront pas côte à côte mais répartis sur une zone. Mais comme les agents se déplacent 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.

Le volume d'émission est à mettre en relation avec le taux d'évaporation et la distance maximale d'émission, il faut qu'ils soient adaptés à la taille de la simulation. Si on pose  $(u_n)_{n \in \mathbb{N}}$  la suite prenant pour valeur le niveau sonore à la case d'émission, on a la relation suivante avec  $v$  le volume émis et  $r$  le taux d'évaporation :  $u_{n+1} = ru_n + v$  ce qui donne pour  $u_0 = 0$  :  $\forall n \in \mathbb{N}, u_n = v(1 - r^n)$  La relation est la même en remplaçant  $v$  par  $v(1 - r^{ds})$  qui prend en

compte l'atténuation selon la distance pour les cases environnantes. Pour éviter que le signal sonore soit propagé à tous les agents, on le limite à 1/5 de la grille autour du centre, donc il se propage en tout sur 2/5 de la grille. Le volume est lui aussi fixé à cette valeur, cela fait que le signal reste plus longtemps sur une grande grille car les agents ont une distance plus grande à parcourir donc ils ont besoin de plus de temps. Le temps d'attente maximal des robots (appelé patience) est lui aussi proportionnel à la taille de la grille pour les mêmes raisons. Il est égal à la moitié de cette taille pour être sûr que les agents proches puissent venir s'ils étaient en chemin mais pour ne pas non plus attendre trop.

## Mécanisme de collaboration – Fusion

Une fois qu'un robot appelant à l'aide a pour voisin un autre robot qui est venu l'aider, les deux fusionnent pour former un MegaRobot. Ce robot est un consensus des deux robots qui peut maintenant déplacer l'objet C. Il pourra alors se déplacer avec l'objet, et agira de la même manière qu'un robot classique, à l'exception près qu'il ne sera pas attiré par les phéromones. Une fois l'objet lâché, il défusionnera et chaque robot reprendra son fonctionnement normal. Lors de la fusion, les deux robots sont mis dans un état inactif et sont retirés de la grille, pour être stockés dans le MegaRobot. À la défusion, le MegaRobot replace les deux robots sur la grille, son état devient inactif et il est retiré de la grille.

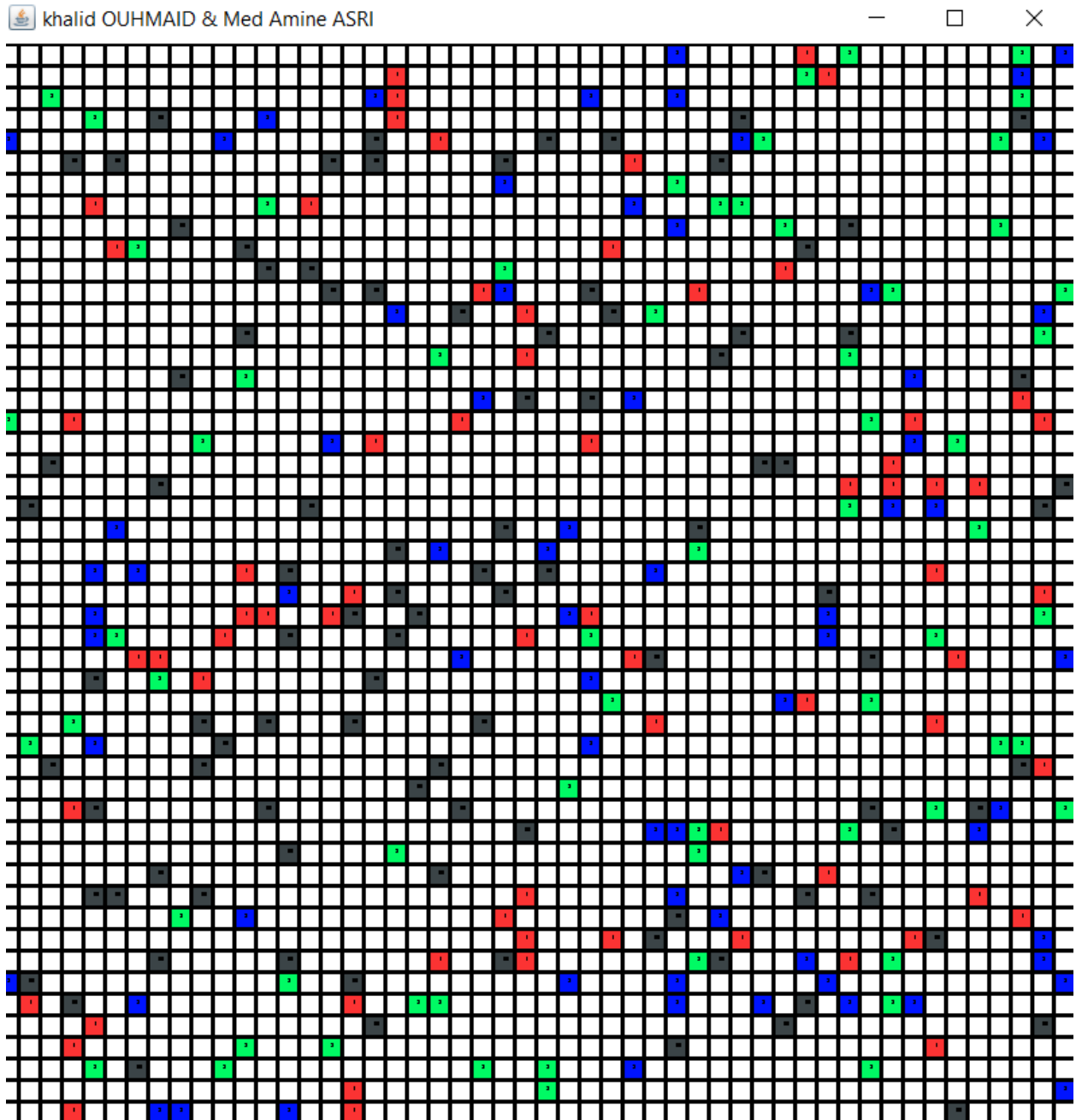


Les cases jaunes représentent des objets de type C. Les croix représentent les robots. Les étoiles représentent les MegaRobots. Les cases vertes représentent les phéromones, l'intensité du vert correspondant à l'intensité de la phéromone. Sur la première image, nos deux robots ont fusionné et ont commencé à déplacer l'objet. Sur la deuxième image, les deux robots ont défusionné après avoir lâché l'objet.

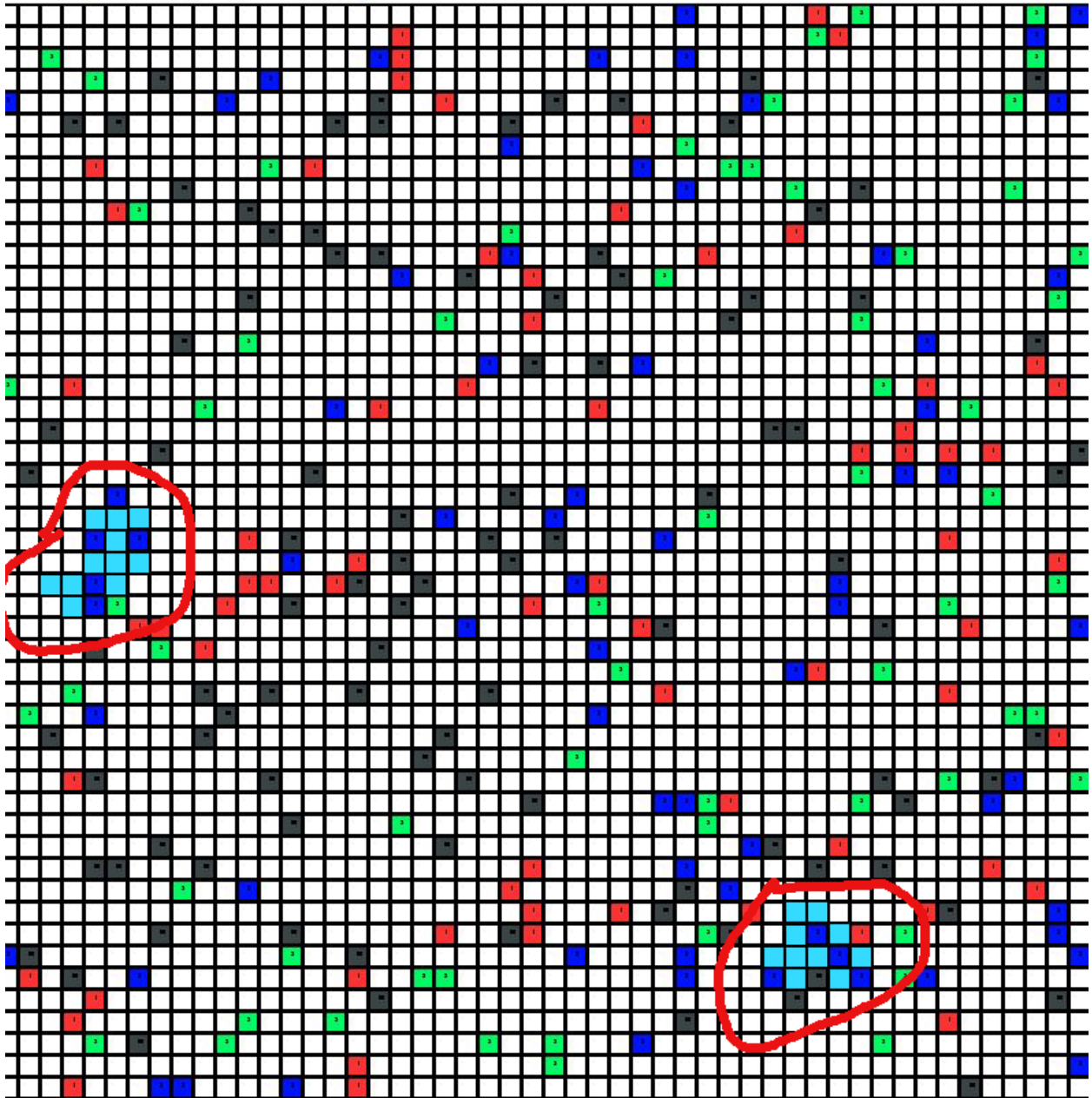


## Résultats obtenus

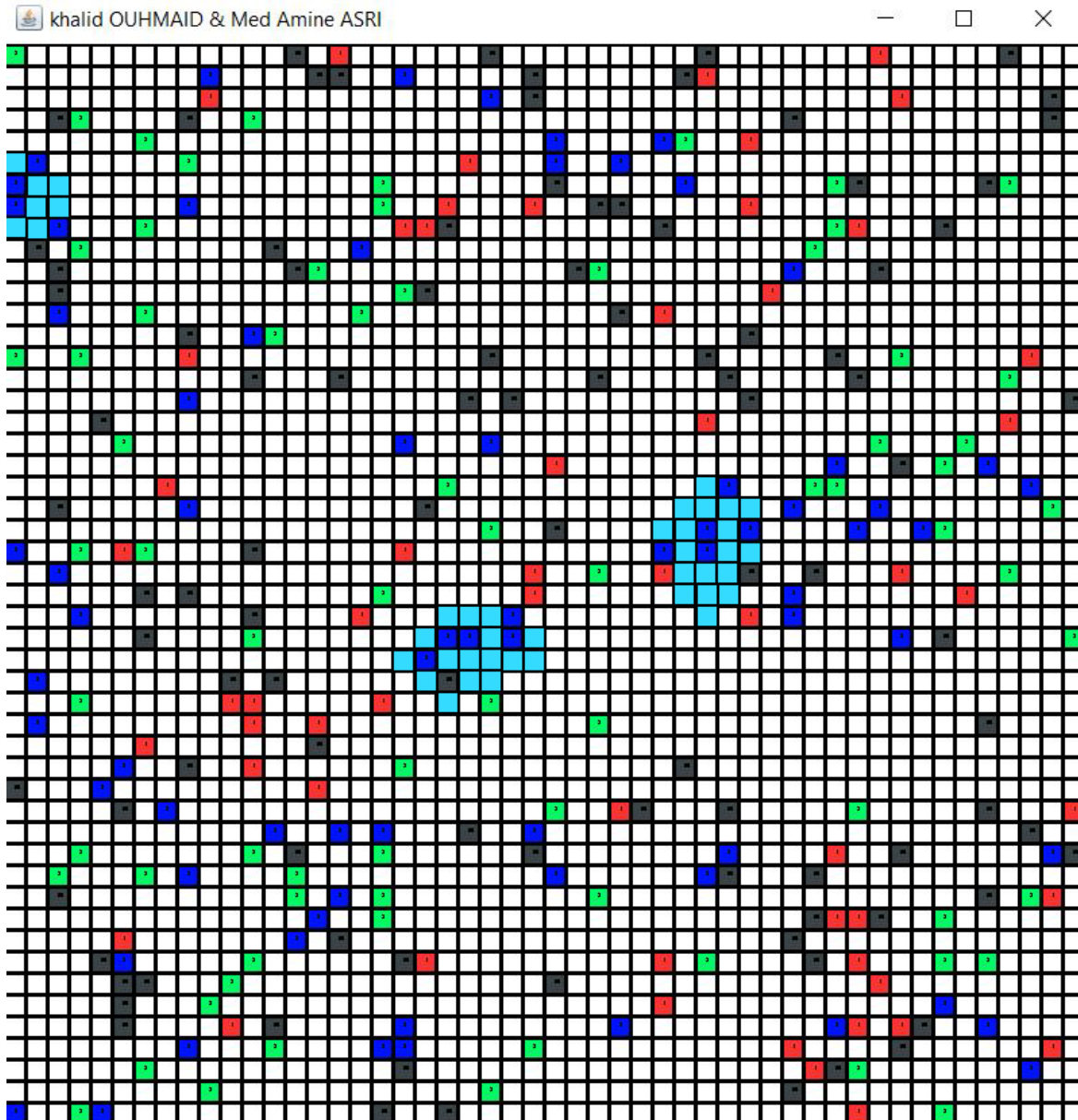
En gardant les paramètres optimaux obtenus à la fin de la partie précédente et en ajoutant 200 objets de type C :



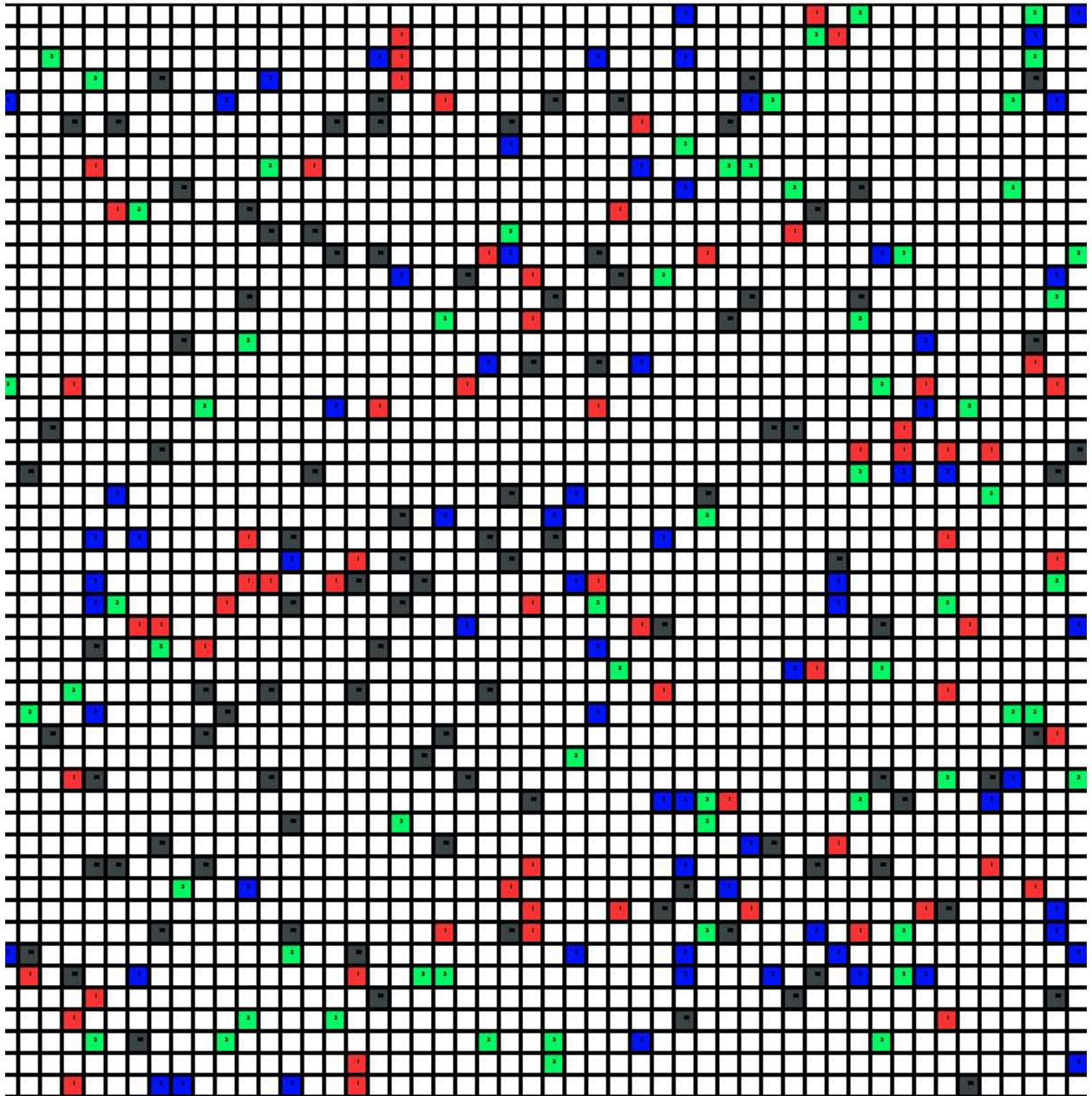
On obtient donc des résultats identiques à ceux de la première partie : avec une mémoire de 1, on a des clusters sans bruit.



On peut observer sur l'image ci-dessus l'effet des phéromones sur les robots, ceux-ci obéissant strictement aux phéromones, une tendance à délaisser massivement les autres tâches apparaît lorsqu'un phéromone est émis. On peut réduire cette tendance grâce au paramètre `chanceToSniff` de la classe `Robot`, qui donne le pourcentage de chance qu'un robot s'intéresse aux phéromones durant une itération. Avec ci-dessous `chanceToSniff = 0.5` contre `chanceToSniff = 1` ci-dessus.



Très longue exécution avec une perception alternative (lorsque qu'un robot porte un objet il ne considère maintenant comme cases voisines que les cases à une distance de manhattan = 1, au lieu de considérer ces dernières + les cases en diagonale)



La video demo\_partie2 a été enregistré avec cette nouvelle perception

## Conclusion

Ainsi, nous venons de réaliser la simulation d'un système multi-agents qui tri collectivement des objets. Ce tri est soit réalisé par un agent seul lorsqu'il s'agit de déplacer un objet A ou B soit réaliser à deux agents lorsqu'il faut déplacer un objet C. Afin de réaliser un tel système, nous nous sommes inspirés du modèle de dépôt de phéromone observable dans la nature. Grâce a cette simulation, on arrive à trier dans l'espace des objets demandant la coopération d'agents en utilisant des agents uniquement réactifs qui émettent et perçoivent un signal sonore mais qui n'ont pas connaissance de leur position, ne peuvent observer que leur environnement proche, et ne communiquent pas entre eux.