

Rapport de projet
Systeme multi-agents

Ce projet consiste à concevoir et implémenter un système multi-agent dont le but est de regrouper des ressources réparties aléatoirement sur une grille, grâce à un comportement émergent des actions des agents. Un système à base de phéromones sera utilisé pour améliorer les performances.

I Le système multi-agent.

(Le code source du premier système est dans le répertoire SMA. Il ne prend pas en compte les phéromone ni le point d'attraction.)

Le système utilise deux matrices distinctes : la première, appelée grille, contient les ressources. La seconde, appelée carte, garde en mémoire les passages des agents, phéromones... et permet de déterminer les probabilités lors du choix de la direction.

La grille est une simple matrice d'entiers. Une valeur de 0 signifie que la case est vide. Sinon, la case contient une ressource dont le type correspond au nombre inscrit.

Lors de la création, elle va générer aléatoirement un nombre spécifié de ressources, puis va les répartir aléatoirement de la manière suivante : on tire les coordonnées au hasard, si la case n'est pas occupée, on pose la ressource, sinon on tire un nouveau jeu de coordonnées.


La fonction voisins compte le nombre de ressources du type spécifié dans les 8 cases entourant la case demandée.

La carte est une matrice de tableaux de taille 12. Les huit premiers éléments permettent de tracer le parcours des agents : lorsqu'un agent passe sur une case, il inscrit la valeur 30 dans le tableau à l'indice correspondant à sa direction. La fonction d'évaporation décrémente cette valeur à chaque itération. On peut ainsi, dans l'interface, tracer les parcours des agents. Les trois éléments suivants correspondent aux phéromones laissées par les agents. Ils permettent de calculer les poids lors du choix de la direction. Le dernier élément détermine la force d'un champ d'attraction sur chaque case.

Les agents ont pour attributs leur position sur la grille, la direction du dernier déplacement (utilisé pour tenir compte de l'inertie) et une variable charge. Comme pour la grille, cette variable définit la ressource transportée : 0 indique que l'agent ne transporte pas de ressource, sinon il transporte une ressource correspondant au nombre. La fonction step() détermine l'action de la prochaine étape : prendre une ressource, poser une ressource ou avancer.

Le choix de la direction :

J'ai définie les direction par des nombres :

3	2	1
4		0
5	6	7

on attribue à chaque case un poids qui dépend de la direction de l'agent (inertie), de la présence de phéromones ou d'un point d'attraction. On incrémente une variable de ce poids, puis on stocke la valeur dans un tableau. On tire ensuite un nombre au hasard entre 0 et la dernière valeur du tableau. On cherche ensuite dans quel intervalle se trouve ce nombre pour déterminer la direction

exemple :

on remplit le tableau :

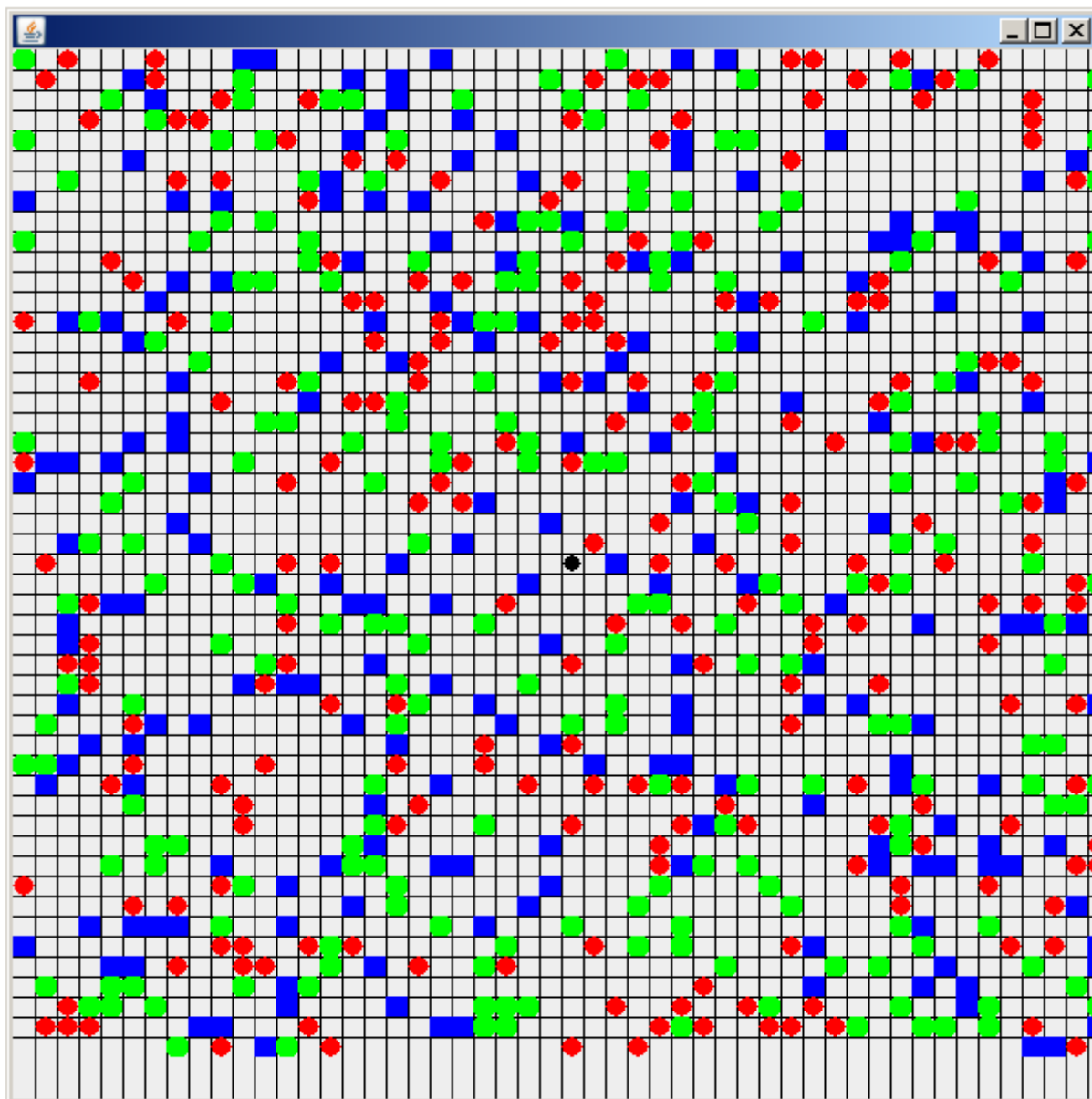
[10;20;130;250;300;310;320;330]

on tire le nombre suivant : 307

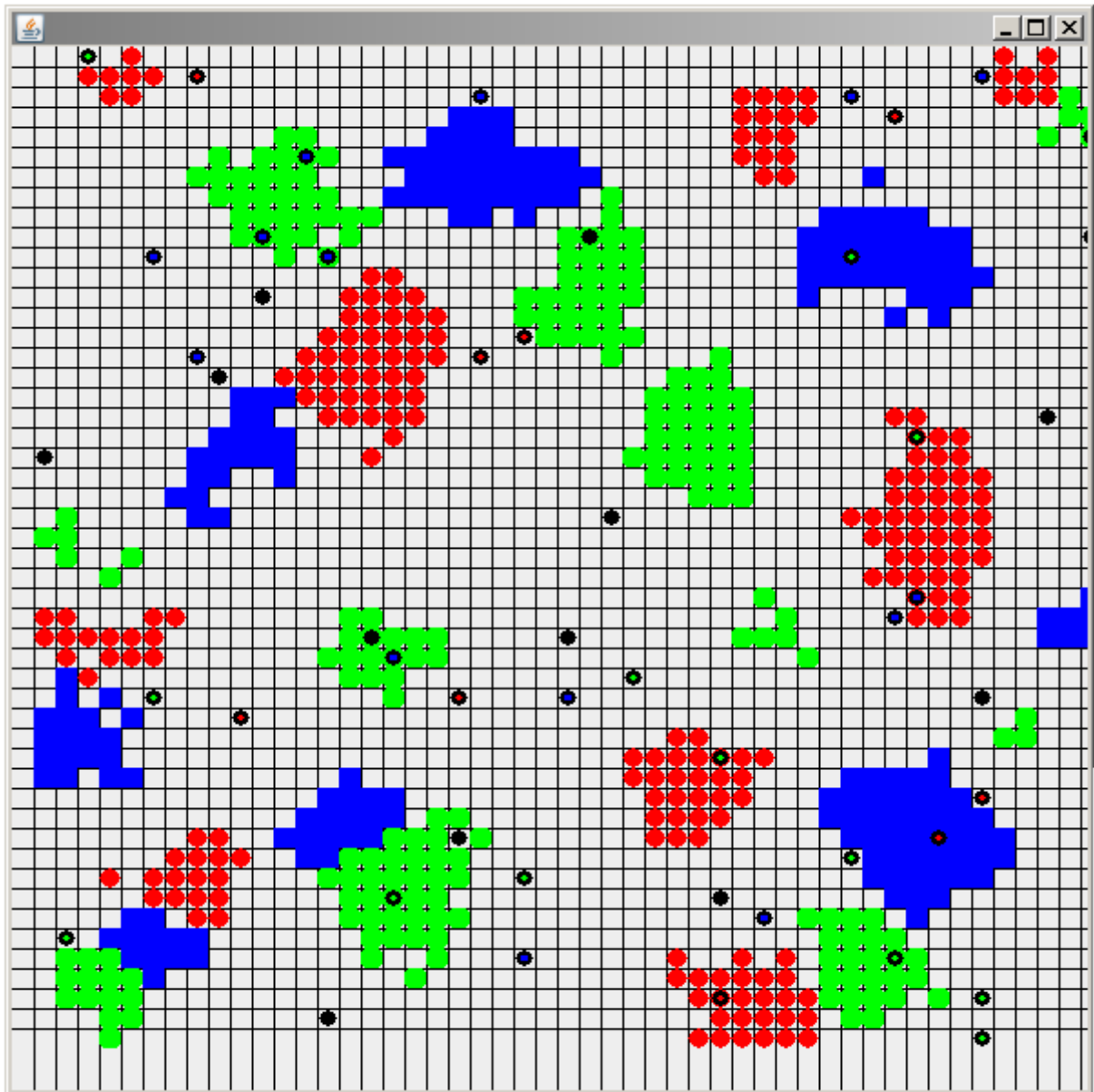
on a $300 < 307 < 320$

La prochaine direction sera donc la direction 5.

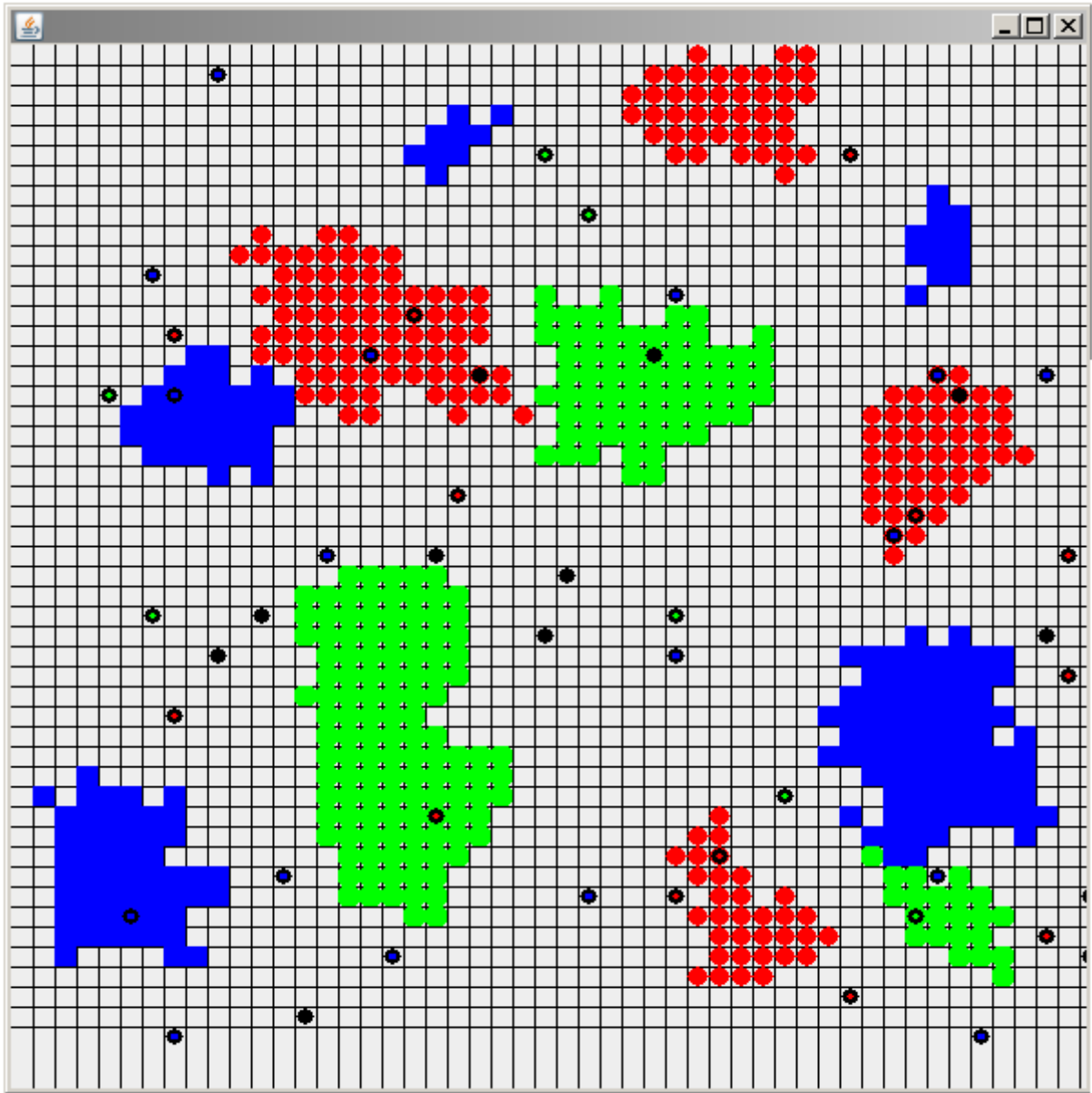
Le premier système affiche les deux matrices : la grille avec les agents et les ressources, la carte et les chemins des agents, puis effectue un million d'itération. Pour plus de rapidité, la mise à jour des affichages ne s'effectuent que toute les 10 itérations.



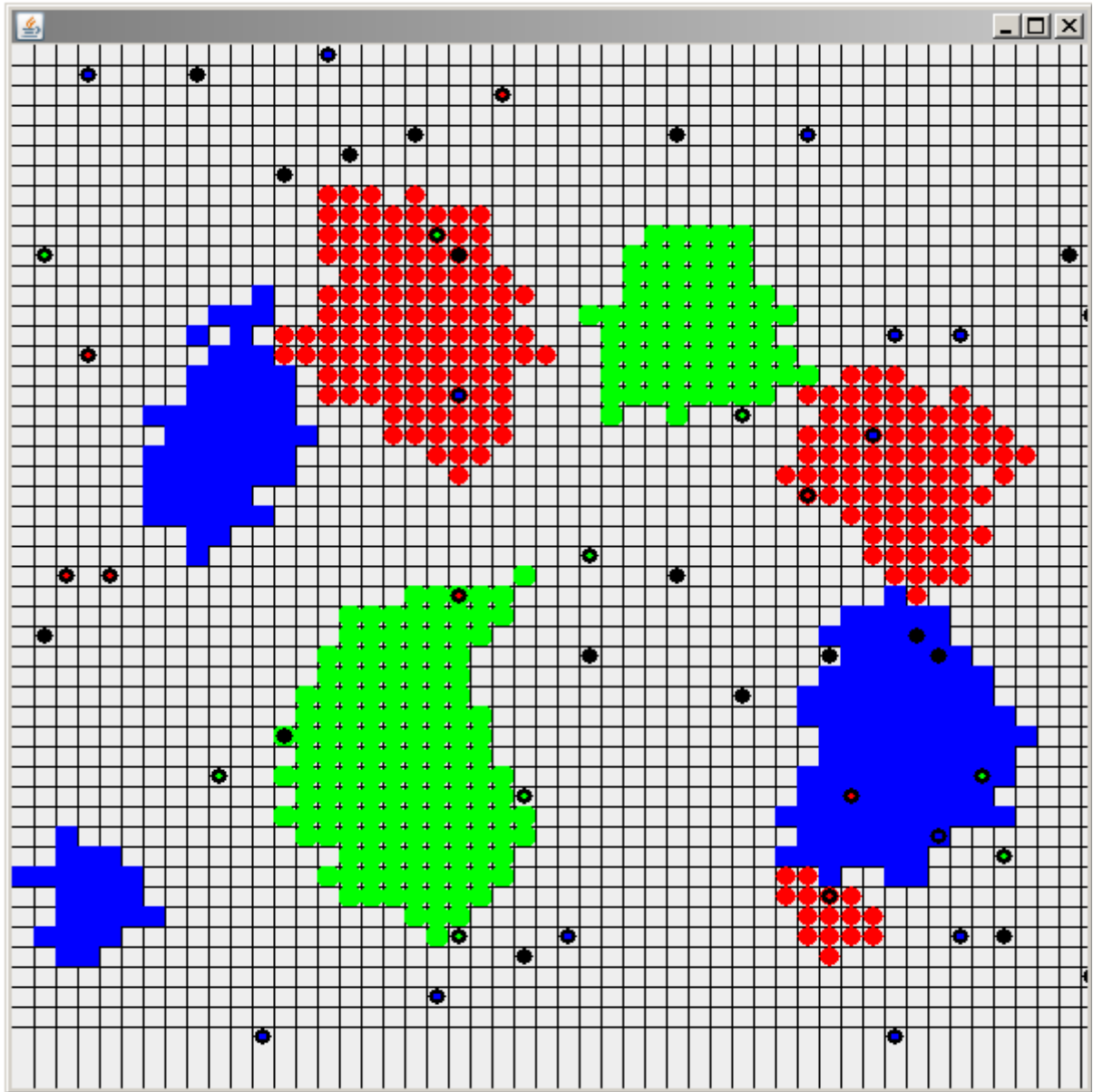
Grille initiale : 650 ressources sont réparties sur une matrice 50X50



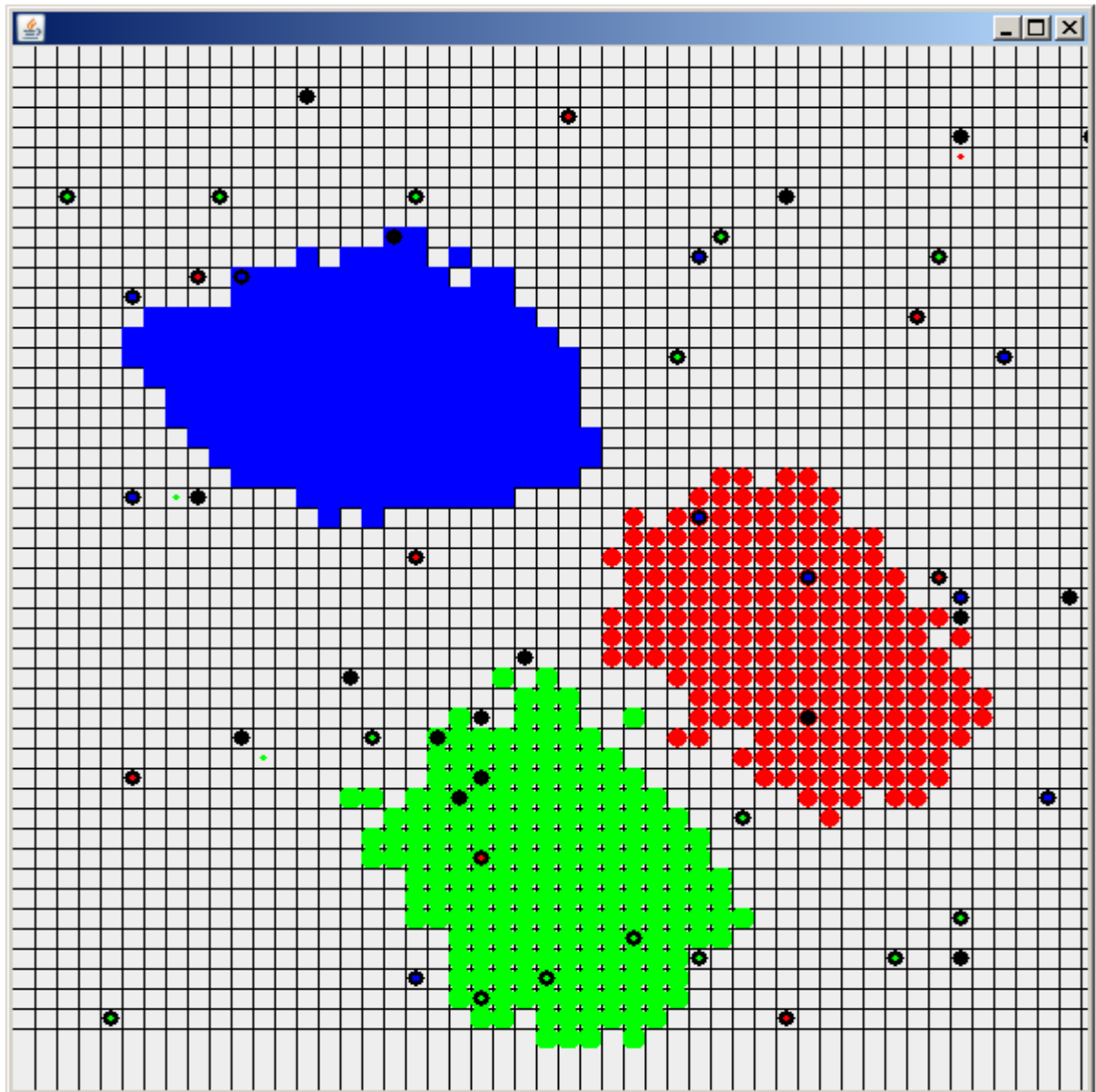
au bout de 50 000 itérations, des tas commencent à émerger



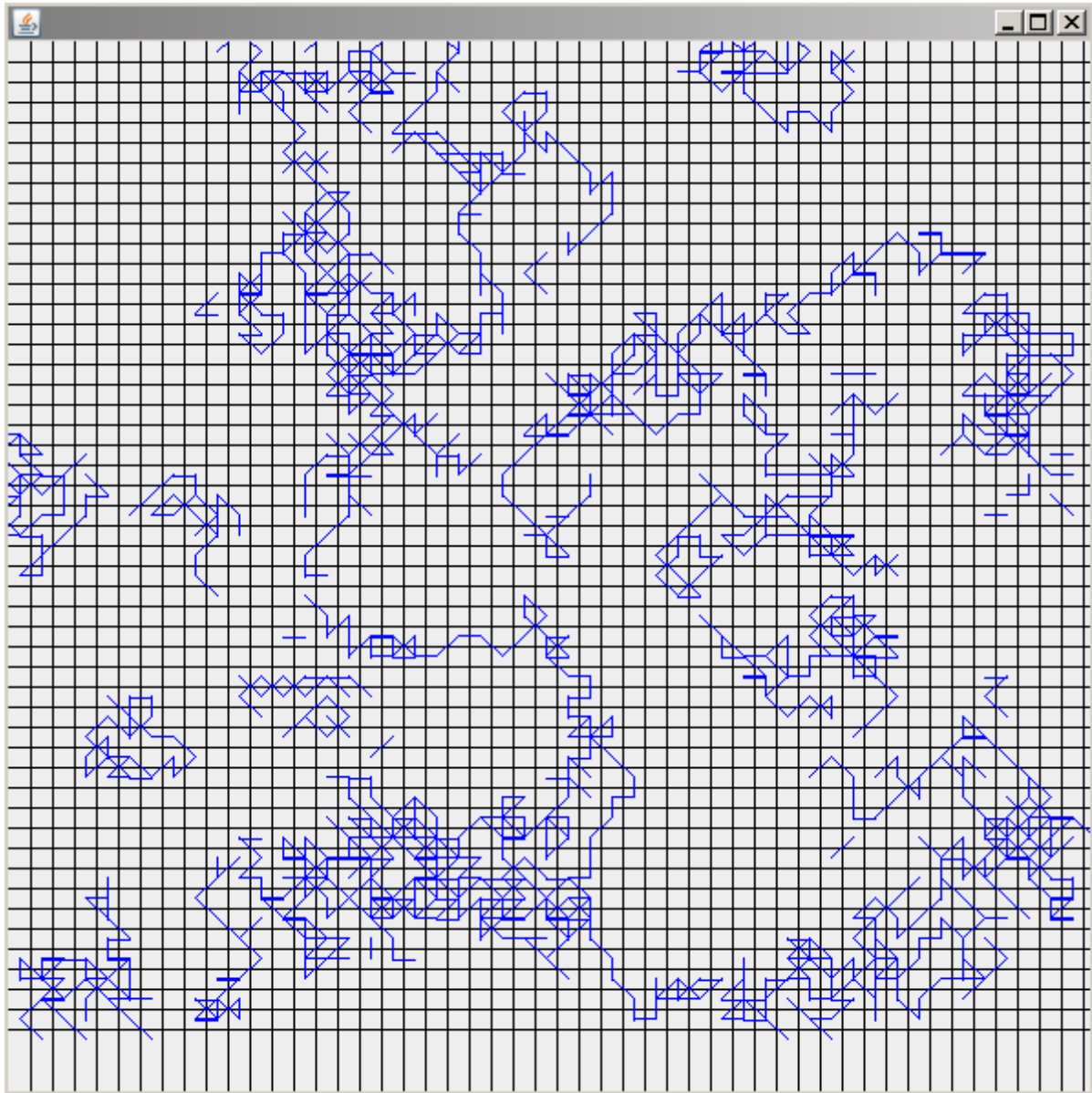
mais le système commence à stagner : au bout de 500 000 itérations, les tas sont plus gros mais restent très nombreux.



Au bout d'un million d'itérations, il reste plusieurs tas de chaque type de ressource.



Il faudra entre 1,5 et 2 millions d'itérations pour obtenir ce résultat



un exemple de carte

On s'aperçoit que le système n'est pas très efficace, nous allons tenter de l'améliorer en s'inspirant d'un système qui a déjà fait ses preuves : les colonies d'insectes.

II Une fourmilière virtuelle

Le code source du deuxième système se trouve dans le répertoire SMA2

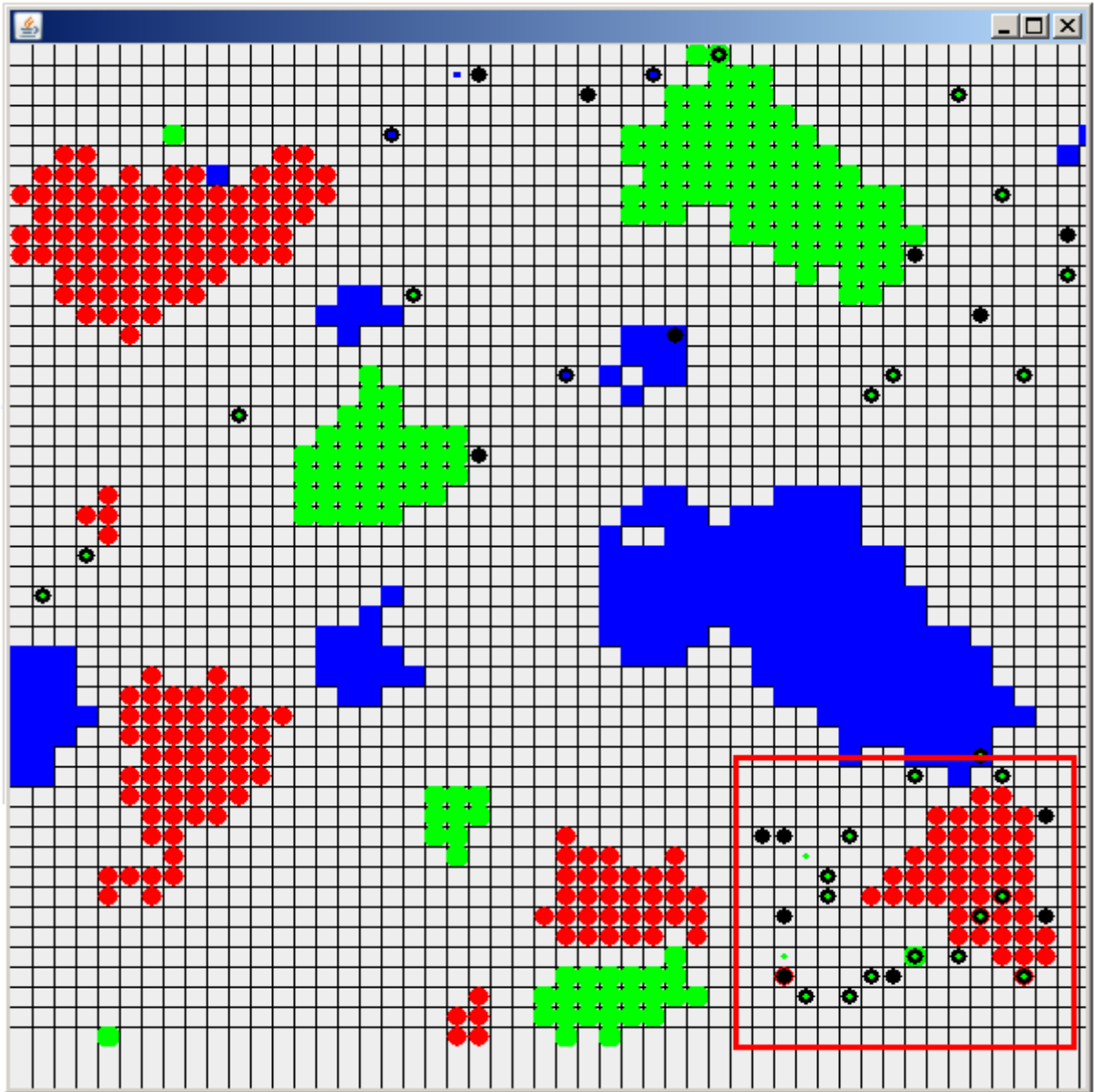
Afin d'améliorer l'efficacité de la recherche des ressources, j'ai implémenté un système à base de phéromones inspiré des fourmis : les agents marquent la carte à leur passage, augmentant la probabilité pour un autre agent de choisir les cases déjà parcourues. Afin d'éviter que les agents ne reviennent sur leur pas trop souvent, seules les trois cases situées devant l'agent sont influencées par les phéromones, les autres cases atteignables gardant une faible probabilité d'être choisies.

Le système repose sur trois phéromones, une pour chaque type de ressource. Les agents à la recherche de ressources ne déposent pas de phéromones, et ne détectent que la phéromone la plus attractive des trois. Ceux qui transportent une ressource déposent la phéromone associée à la ressource, et ne détectent uniquement que cette phéromone. Ainsi, des chemins se forment entre les tas de ressources identiques.

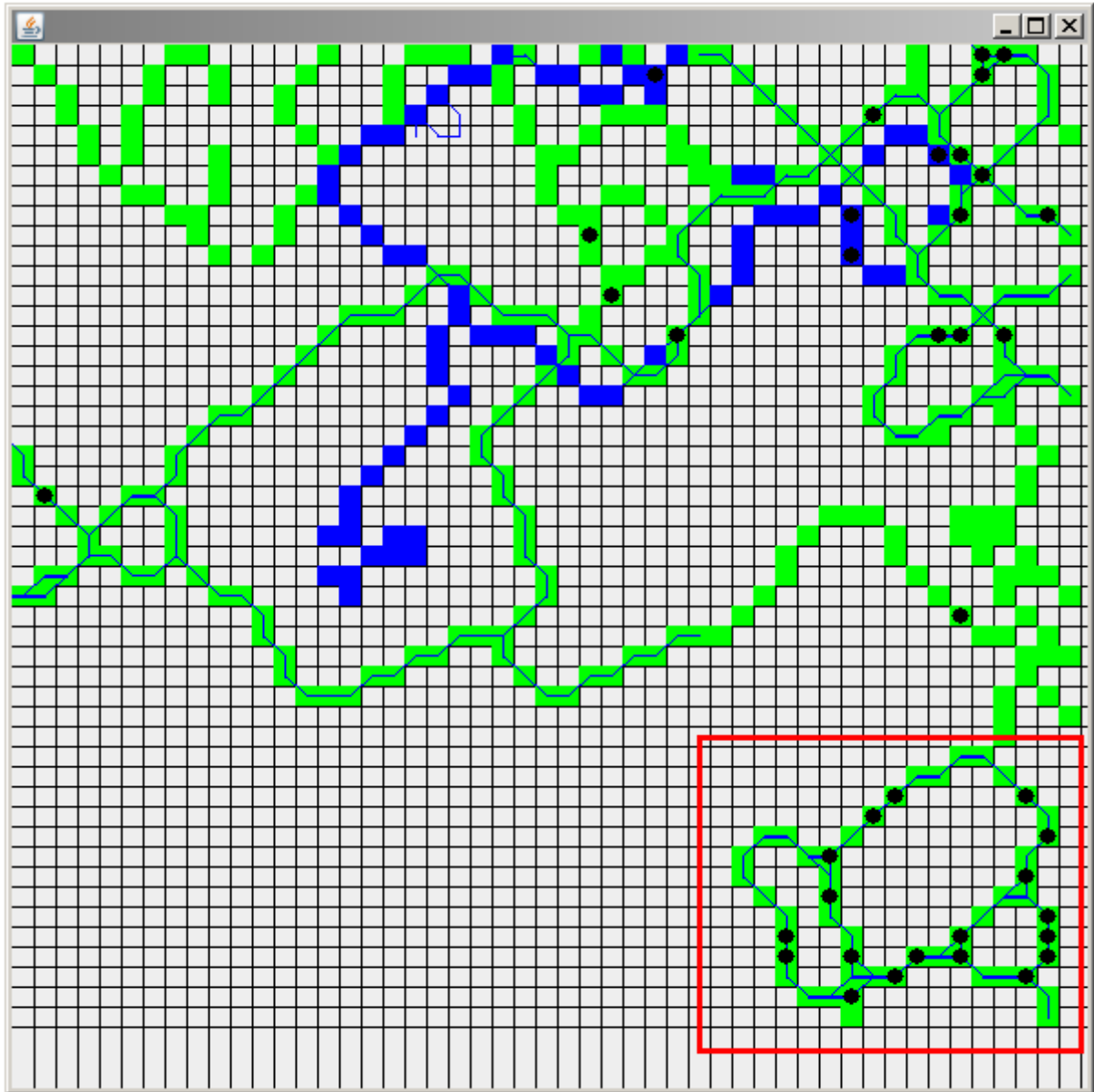
Pour éviter la saturation de phéromones, le marquage des cases ne s'effectue que si la force de la phéromone de la case ne dépasse pas déjà cinq fois la valeur que l'on veut marquer.

Un système d'évaporation permet à chaque étape de diminuer la force des phéromones, permettant de faire disparaître les chemins peu empruntés.

Après quelques essais, j'ai pu me rendre compte que si des chemins se formaient bien entre les tas de ressources, il y avait souvent des cas où les chemins bouclent sur eux-mêmes, les agents qui l'empruntent se retrouvent piégés et continuent de renforcer la piste qui devient de plus en plus attractive.

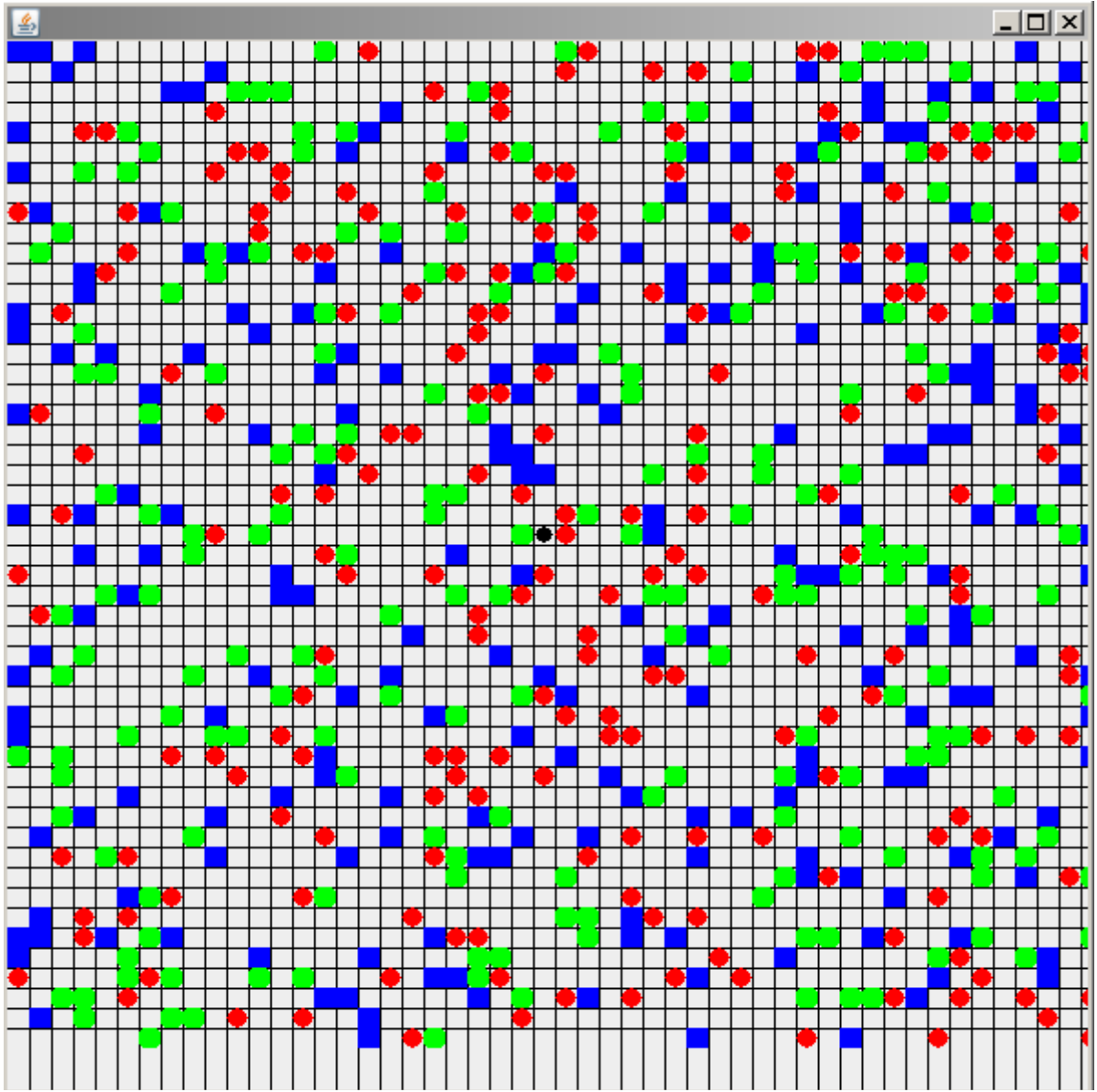


Le piège est ici parfaitement visible sur la grille : les agents tournent en rond

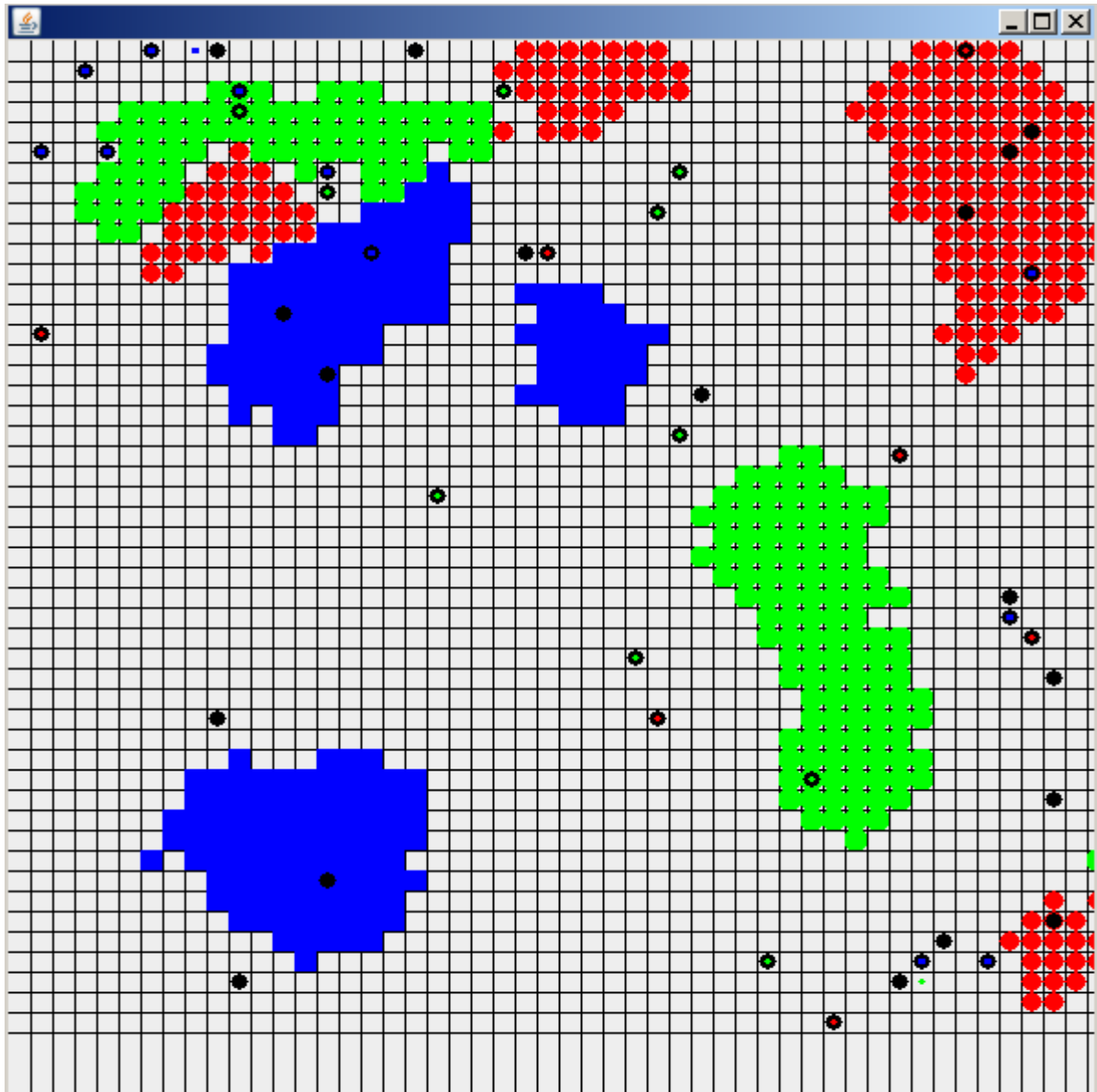


sur la carte, on peut voir parmi les pistes, une boucle : les agents restent prisonnier de la piste qu'ils ont eux-même tracé.

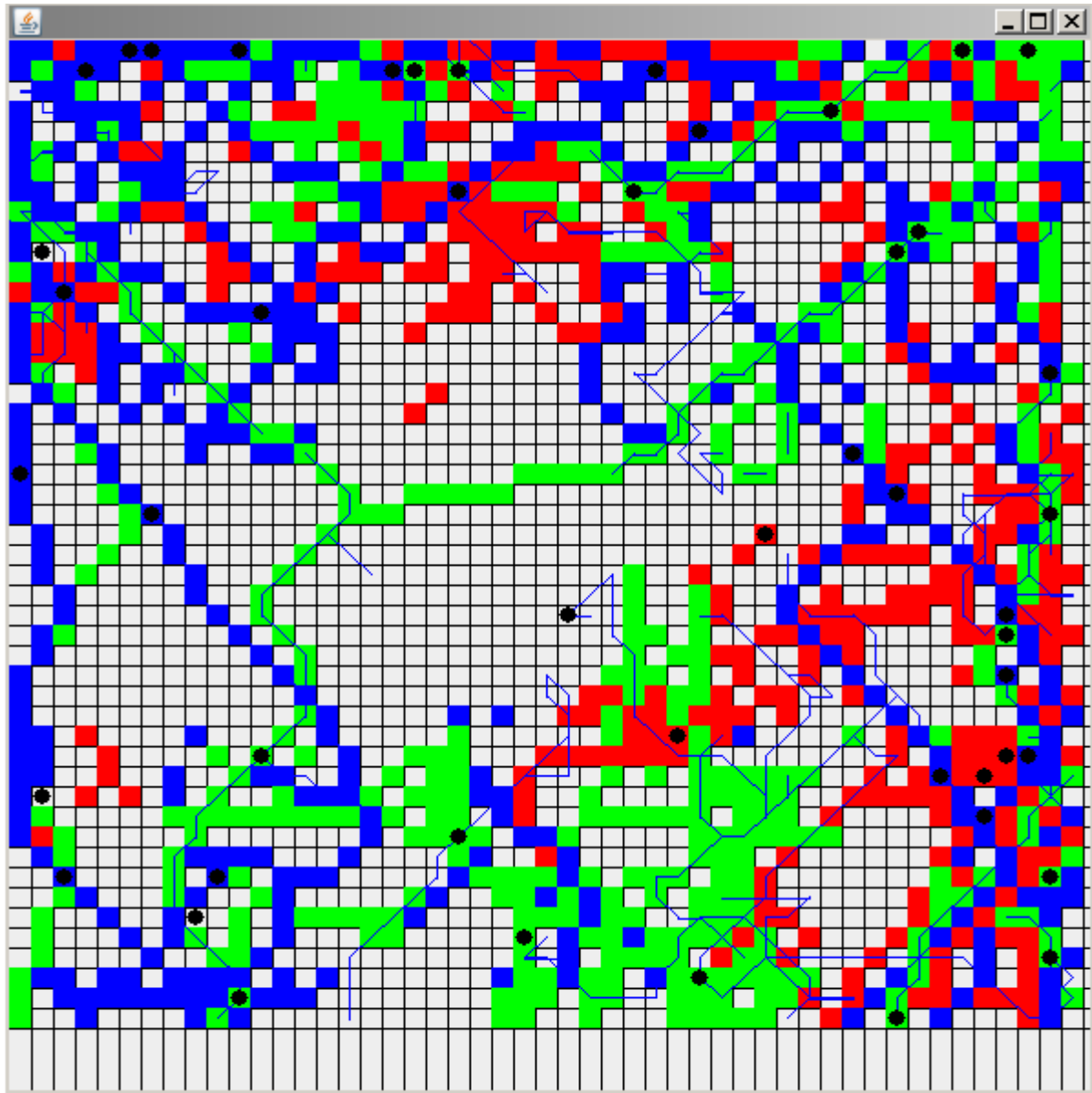
Pour régler ce problème, je me suis encore inspiré des fourmis : de nombreuses espèces utilisent une boussole interne ou la polarité des rayons du soleil pour se guider et éviter de tourner en rond. Il en sera de même pour les agents : Lors de l'initialisation du système, on attribut de façon aléatoire à chaque agent une direction privilégiée. La probabilité d'atteindre la case du coté opposé à cette direction est nulle : l'agent ne peut plus tourner en rond. Cette direction privilégiée ne peut être changée que dans deux situations : si l'agent atteint un bord de la grille, ou lorsqu'il prend ou pose une ressource. Une direction privilégiée est alors attribué au hasard parmi les trois directions opposées à la précédente. L'agent fait ainsi demi-tour et peut tracer (ou suivre) un chemin de retour.



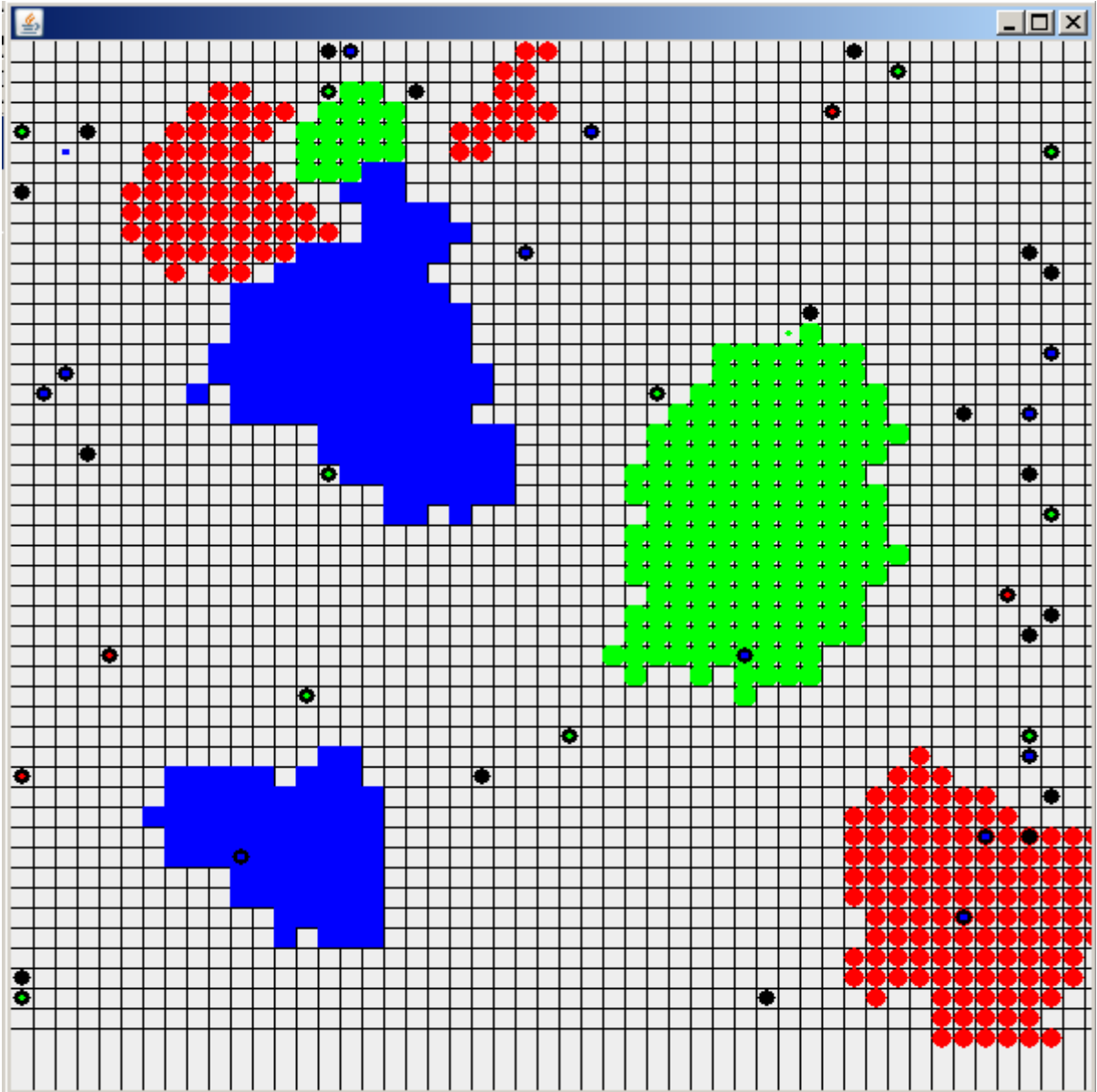
La grille initiale



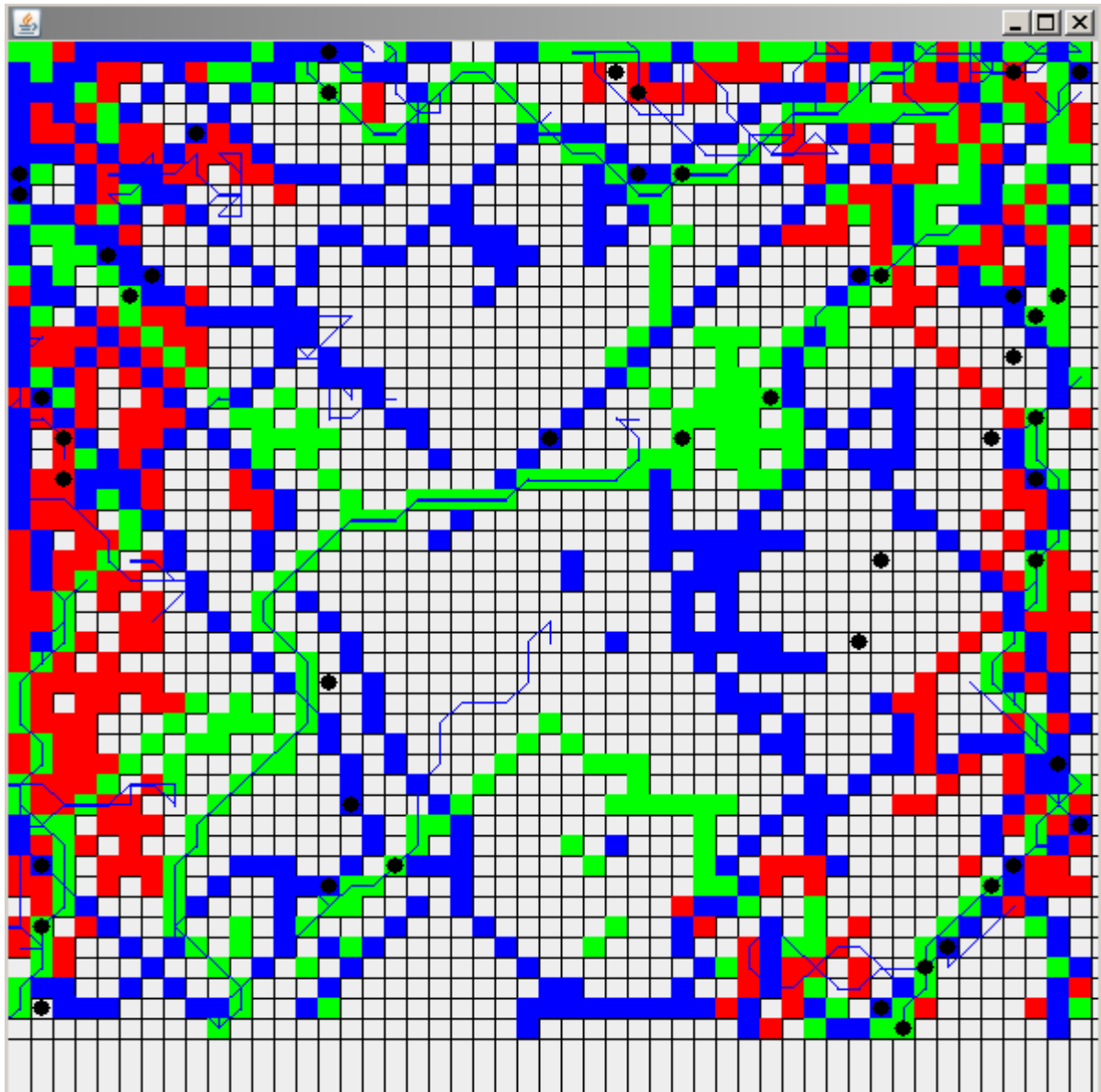
au bout de 50 000 itérations, les tas sont déjà peut nombreux et de taille importante...



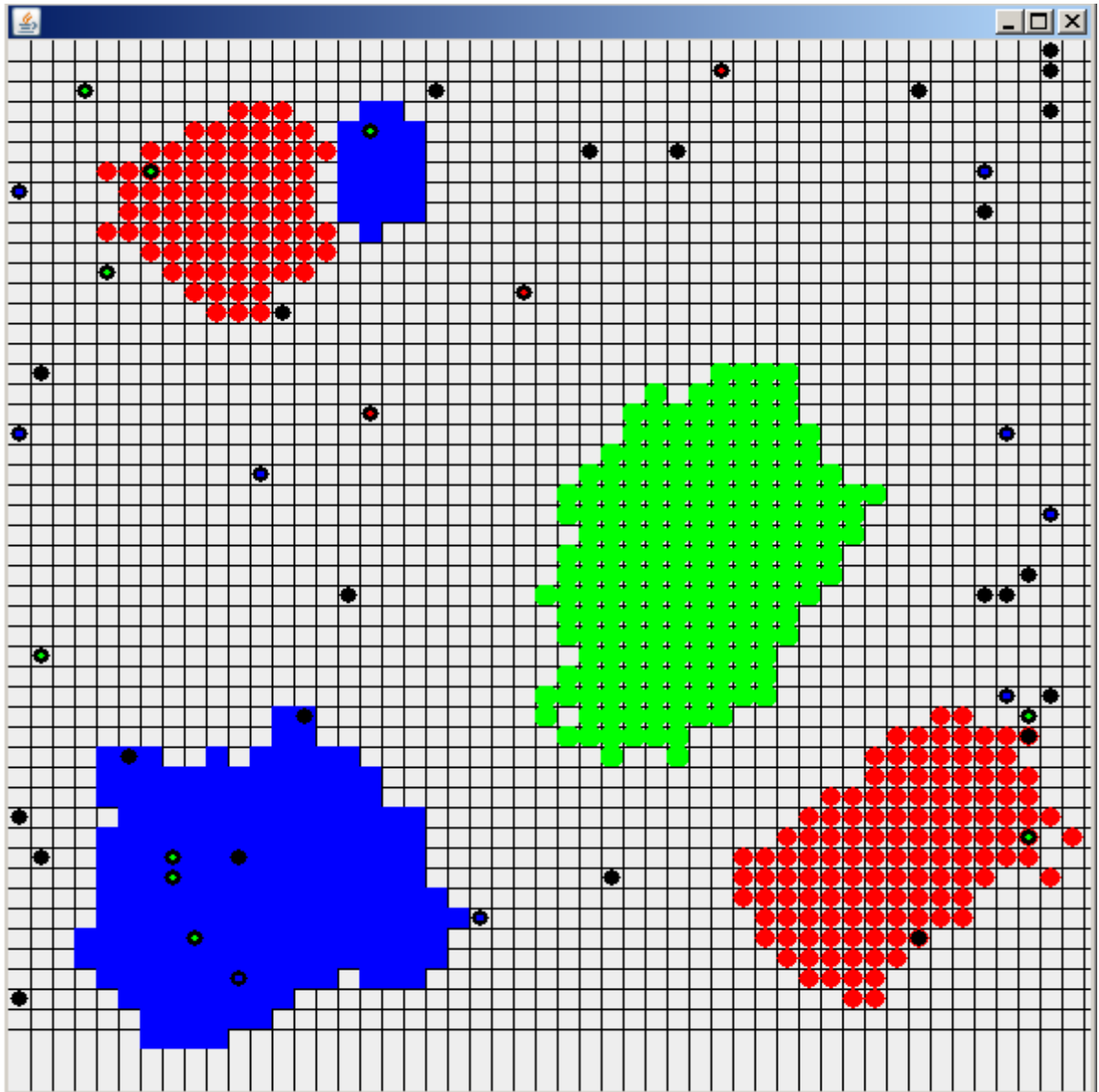
...et des chemins sont déjà bien visibles



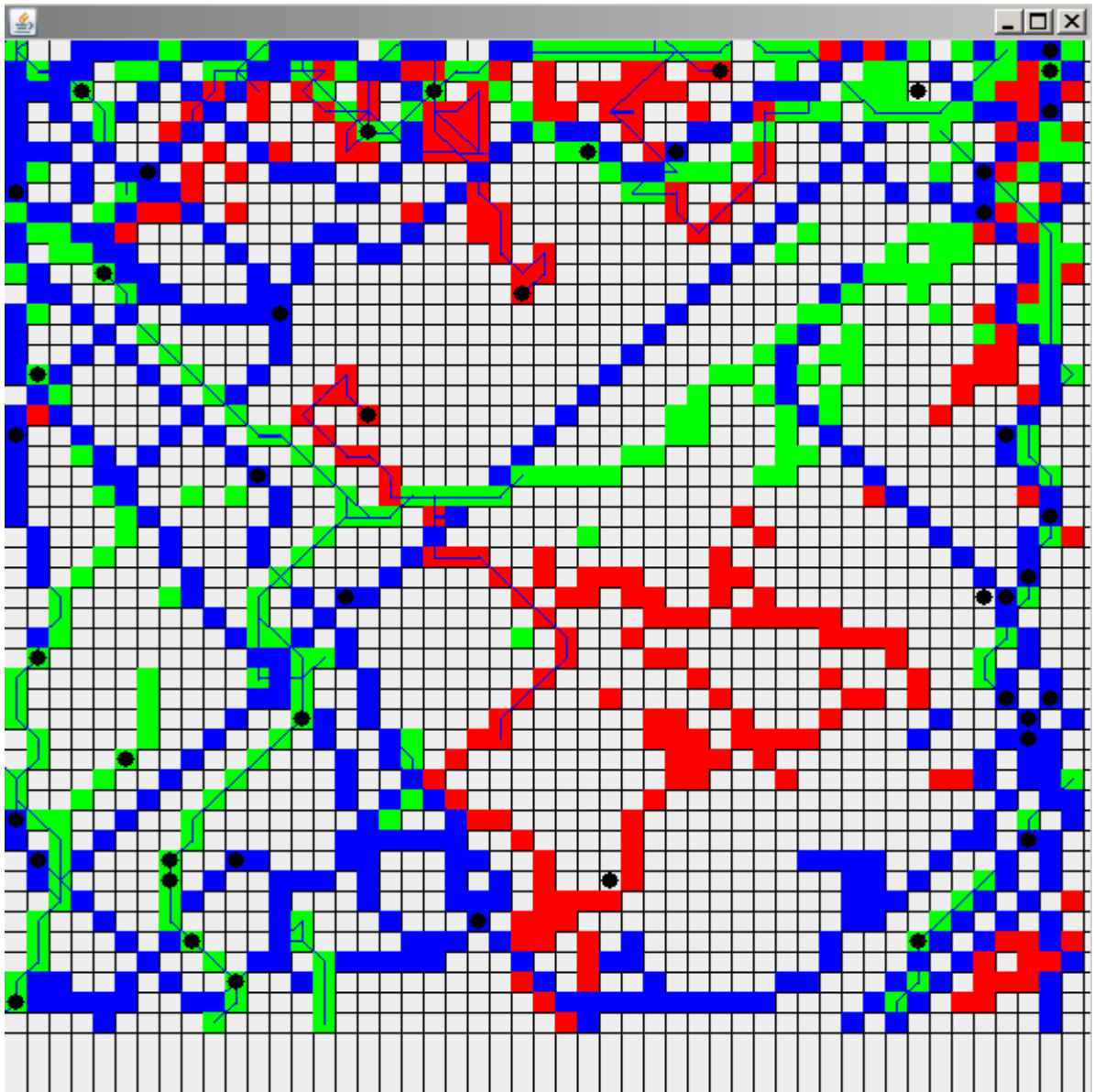
mais au bout de 500 000 itérations, on s'aperçoit que le système n'évolue que très lentement



de même pour les chemin : on reconnait les tracés principaux de la carte précédente



finalement, au bout d'un million d'itérations, les ressources sont relativement bien regroupées, mais pas totalement.



Les chemins n'ont que très peu évolués, cela laisse peut de place à l'exploration.

Malgré cela, le nouveau système obtenu n'est pas beaucoup plus efficace que le précédent, dans certains cas il est même plus lent à cause du calcul des probabilités.

Il y a plusieurs explications à cela :

-Les ressources sont dispersées et il n'y a pas de destination prédéfinie pour les ressources : en effet, les fourmis transportent les ressources d'un tas de ressources vers leur fourmière. Une fois le chemin tracé, le transport est unidirectionnel et unique. Dans le cas des agents, il ne peut y avoir un chemin unique, puisque les ressources sont dispersées, mais un réseau de chemins entre des petits tas qui émergent, et le transport est bidirectionnel puisqu'il n'y a pas de destination. Il faut donc un grand déséquilibre du nombre de ressources pour qu'un tas soit absorbé par un autre.

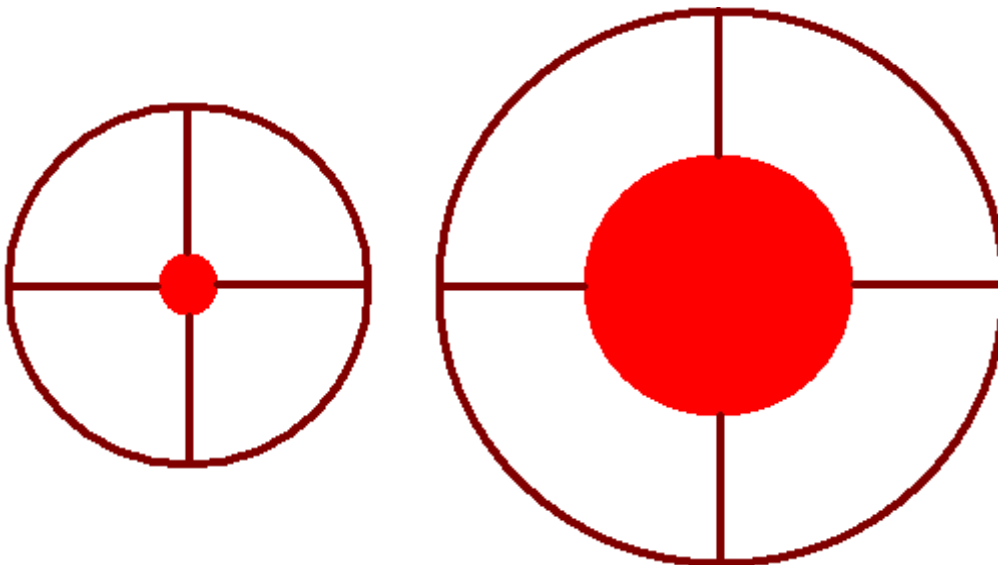
-Une fois que le réseau de chemins est tracé, les agents les suivent et ne s'en écartent que très rarement. Les ressources qui n'ont pas été découvertes dès le début seront délaissées pendant de longues durées. Les fourmis connaissent aussi ce genre de problème, mais il est rare qu'une nouvelle ressource apparaisse à proximité de la fourmilière une fois un chemin établi.

III Un nouveau mécanisme de recherche

La nouvelle idée consiste à augmenter la probabilité qu'un agent passe sur les petits tas sans diminuer la probabilité de déposer une ressource sur un gros tas.

Le système utilisé est le suivant : lorsqu'un agent prend une ressource, il fait demi-tour en marquant les n premières cases de son parcours, en diminuant la force des phéromones déposés à chaque case parcourue. (j'ai choisi $n = \text{taille}/4$). Les agents transportant une ressource ne seront alors plus attirés par les phéromones, il aura donc plus de chance de passer sur un gros tas. Les autres agents à la recherche de ressources seront attirés par ces pistes.

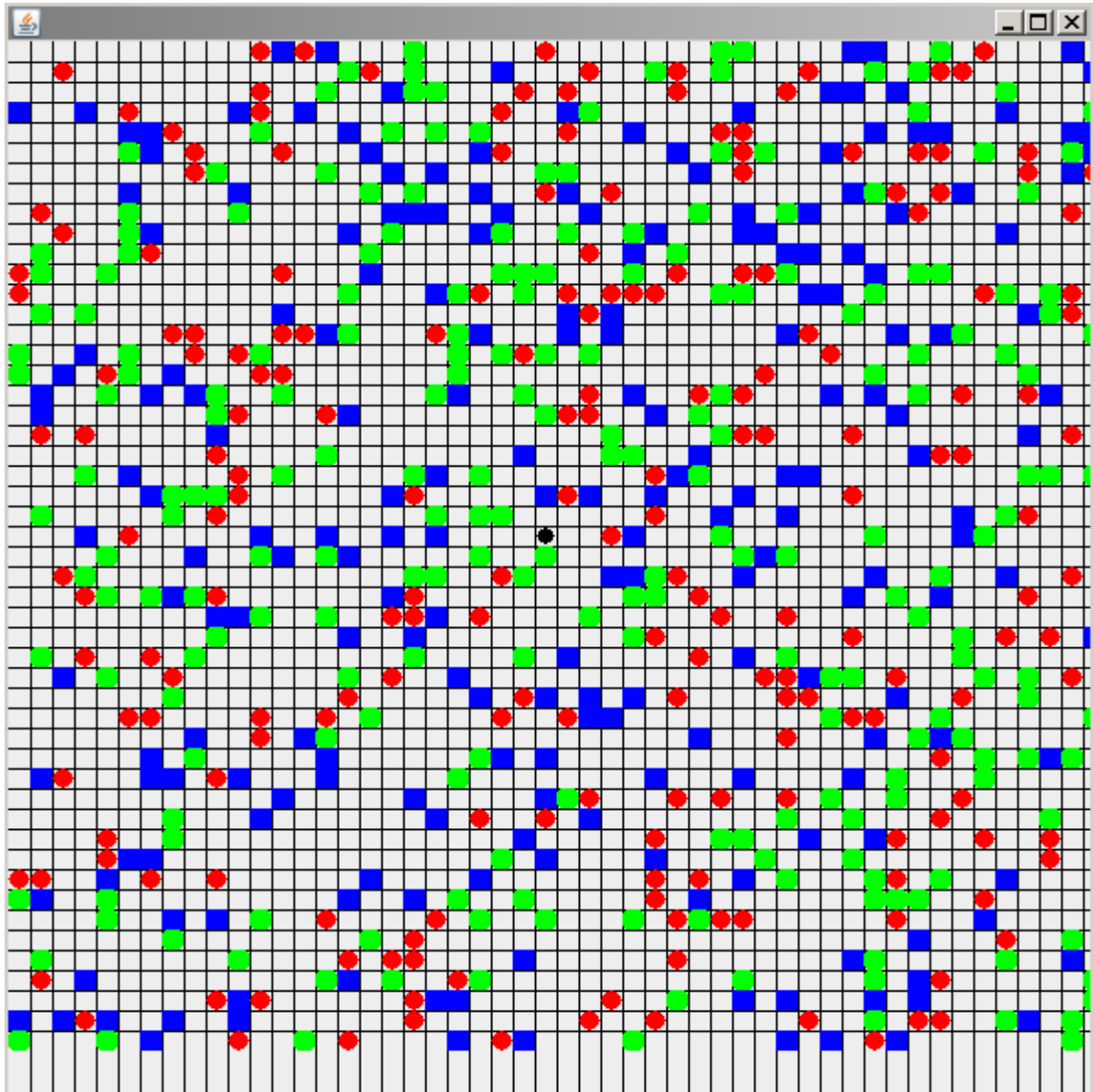
Ce système repose sur le fait que, comme la longueur des pistes est toujours la même, le disque d'influence des grands tas ne sera pas beaucoup plus grand que le tas lui-même. En revanche, celui des petits tas sera considérablement plus grand. Comme les agents transportant des ressources ne détectent pas ces pistes, la probabilité qu'il déposent la ressource sur un tas dépend alors uniquement de la taille de celui-ci.



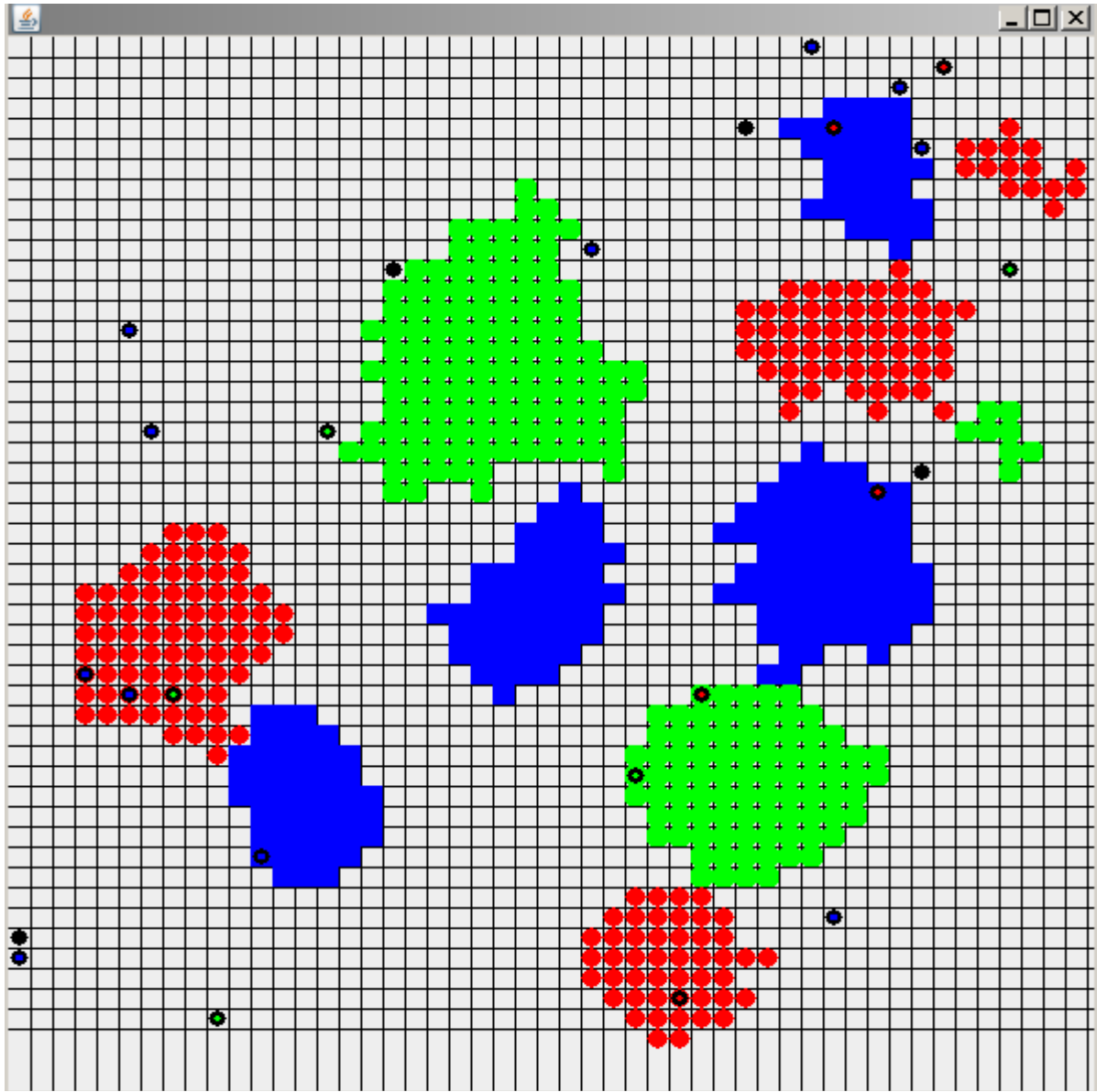
Dans cet exemple, les tas (en rouge) sont de taille très différentes, mais leur disque d'influence sont de même ordre de grandeur. Il sera donc plus simple de prendre une ressource sur un tas de petite taille que d'y en déposer une.

Le système obtenu est alors bien plus rapide, les grands tas proches ont tendance à se rapprocher jusqu'à fusionner tandis que les petits tas se font absorber en dessous d'une certaine taille.

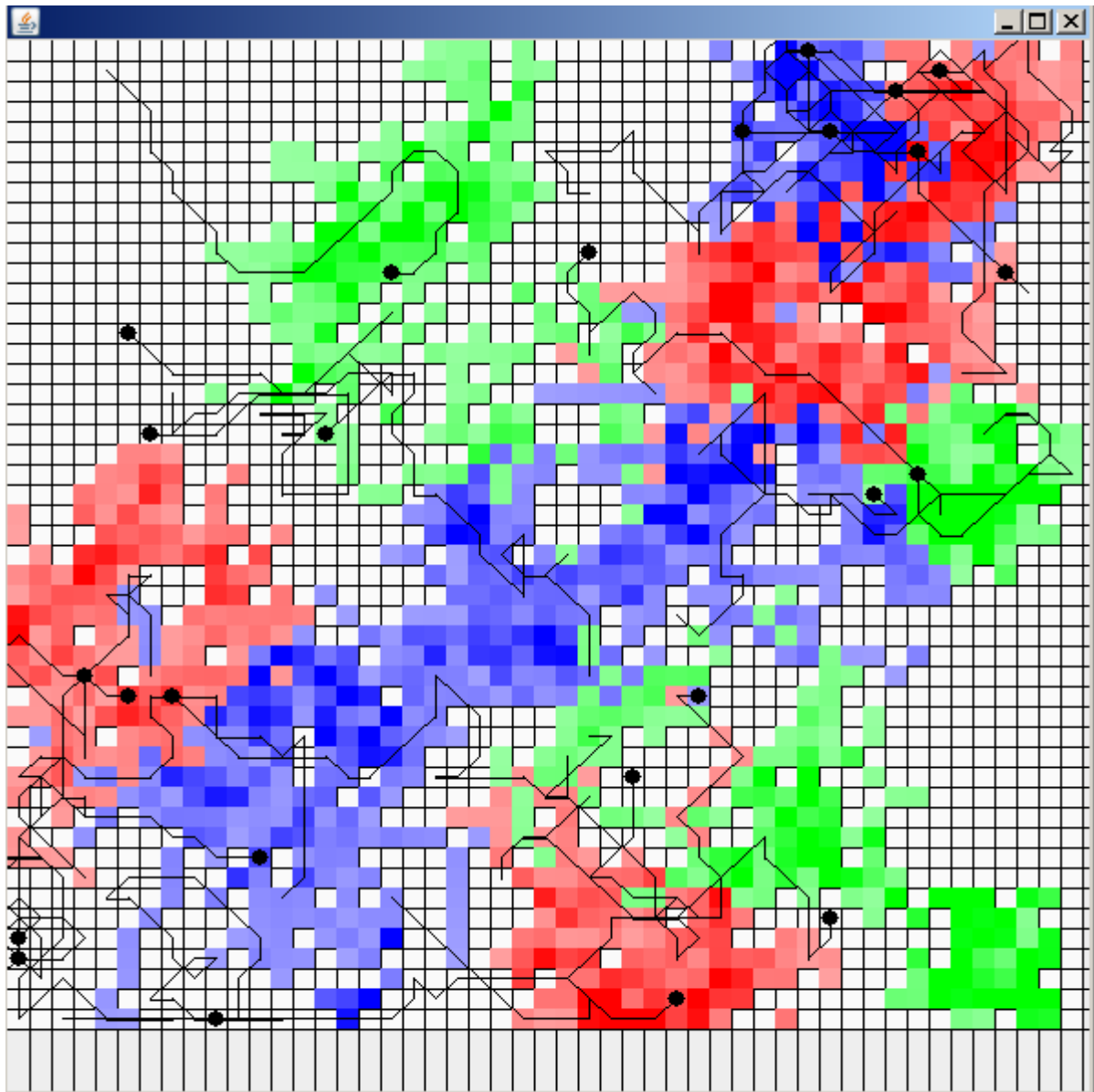
Voici l'exécution de ce système :



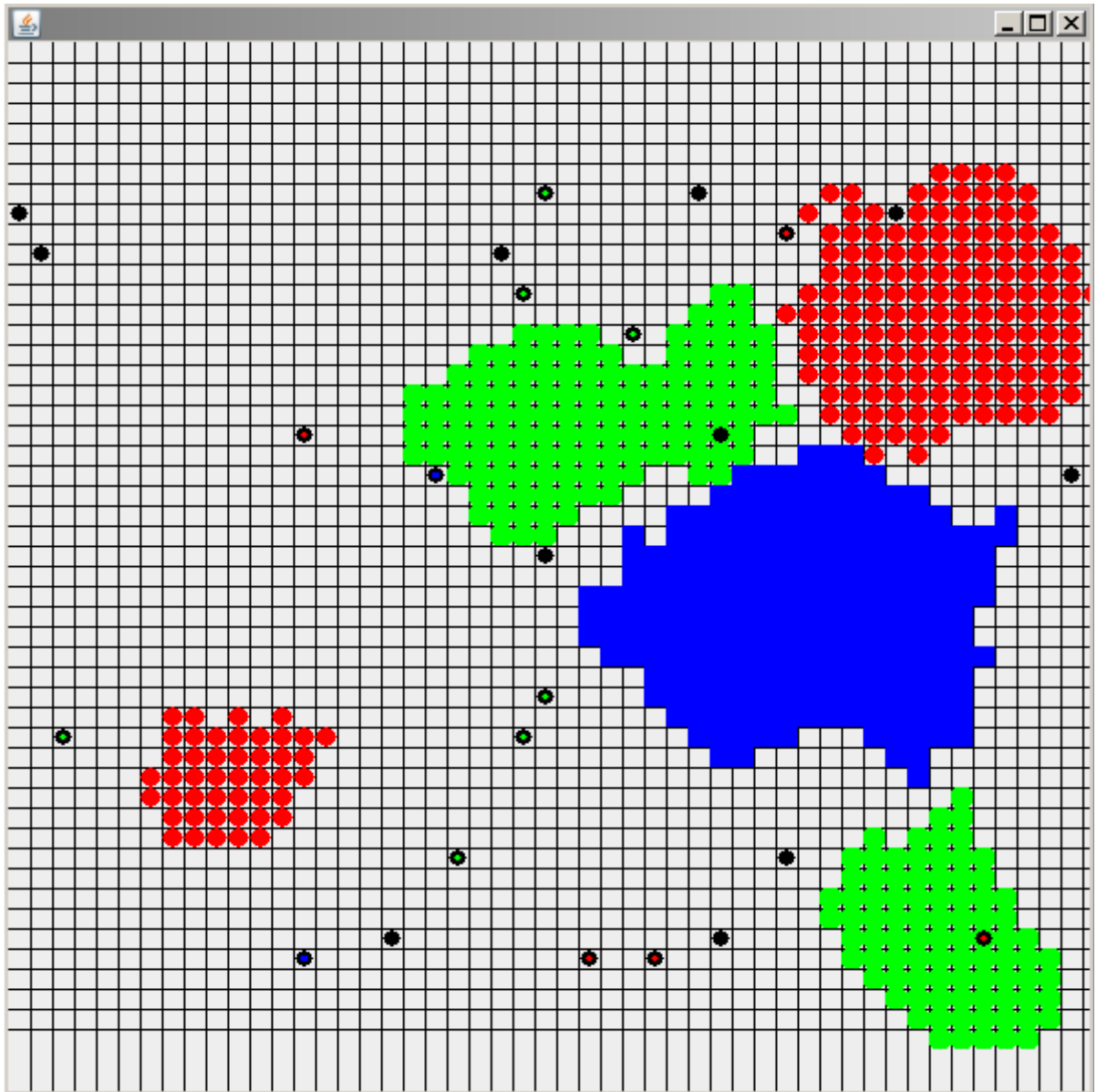
Grille initiale



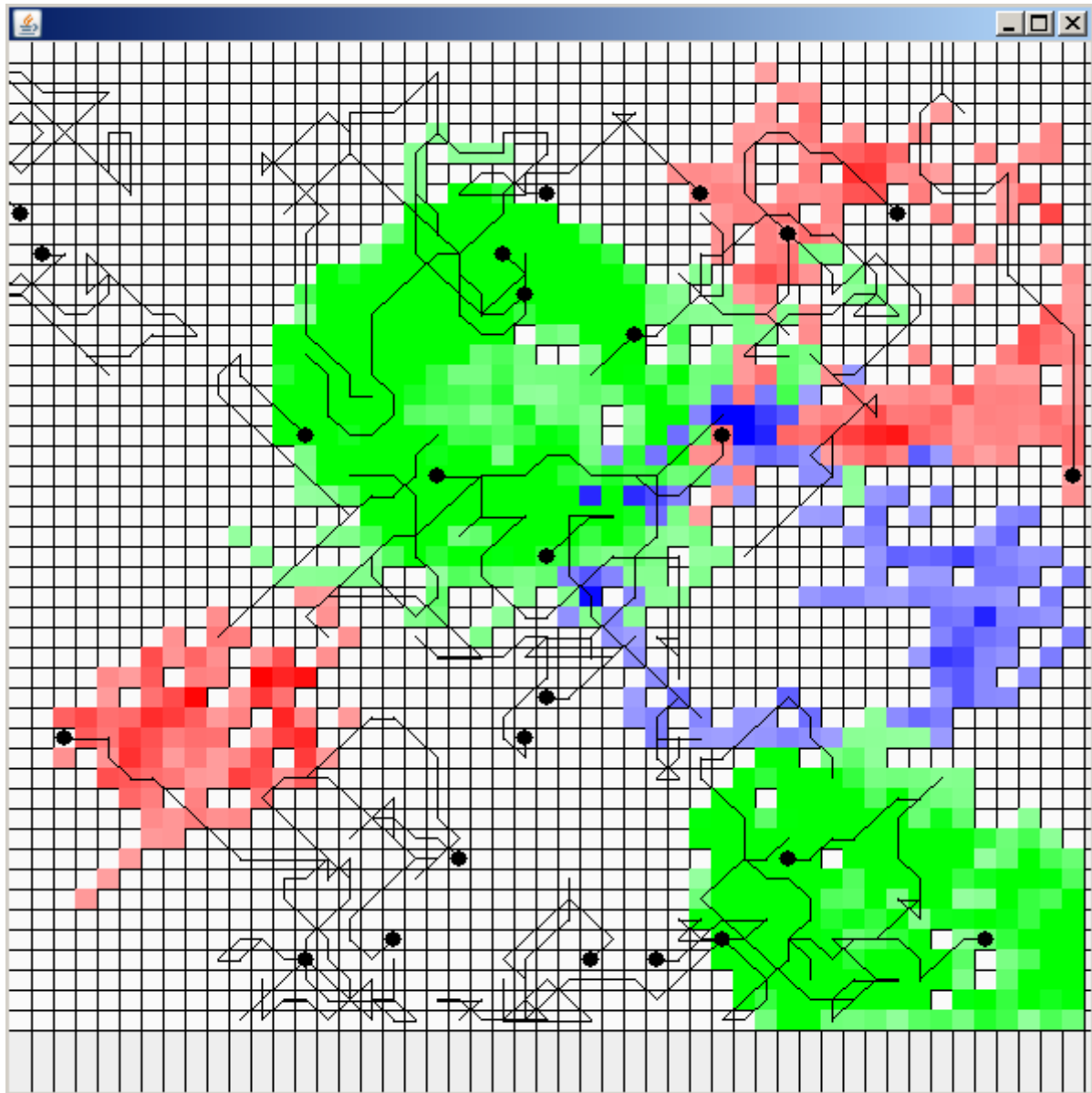
au bout de 50 000 itérations, les ressources sont déjà bien regroupées



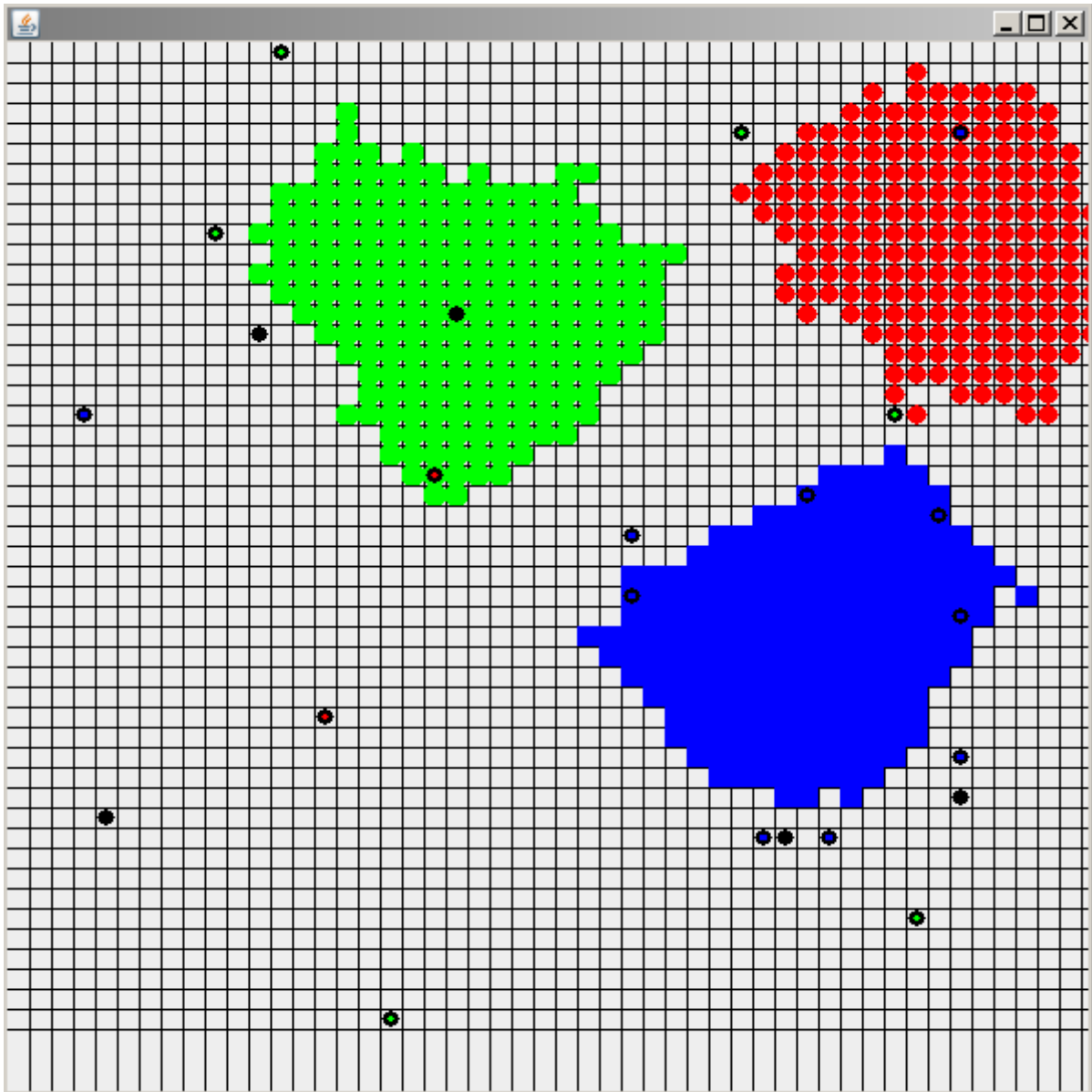
on reconnaît bien les disques d'influence (le tas vert en bas à droite a disparu juste avant le clic, il reste encore quelques phéromones)



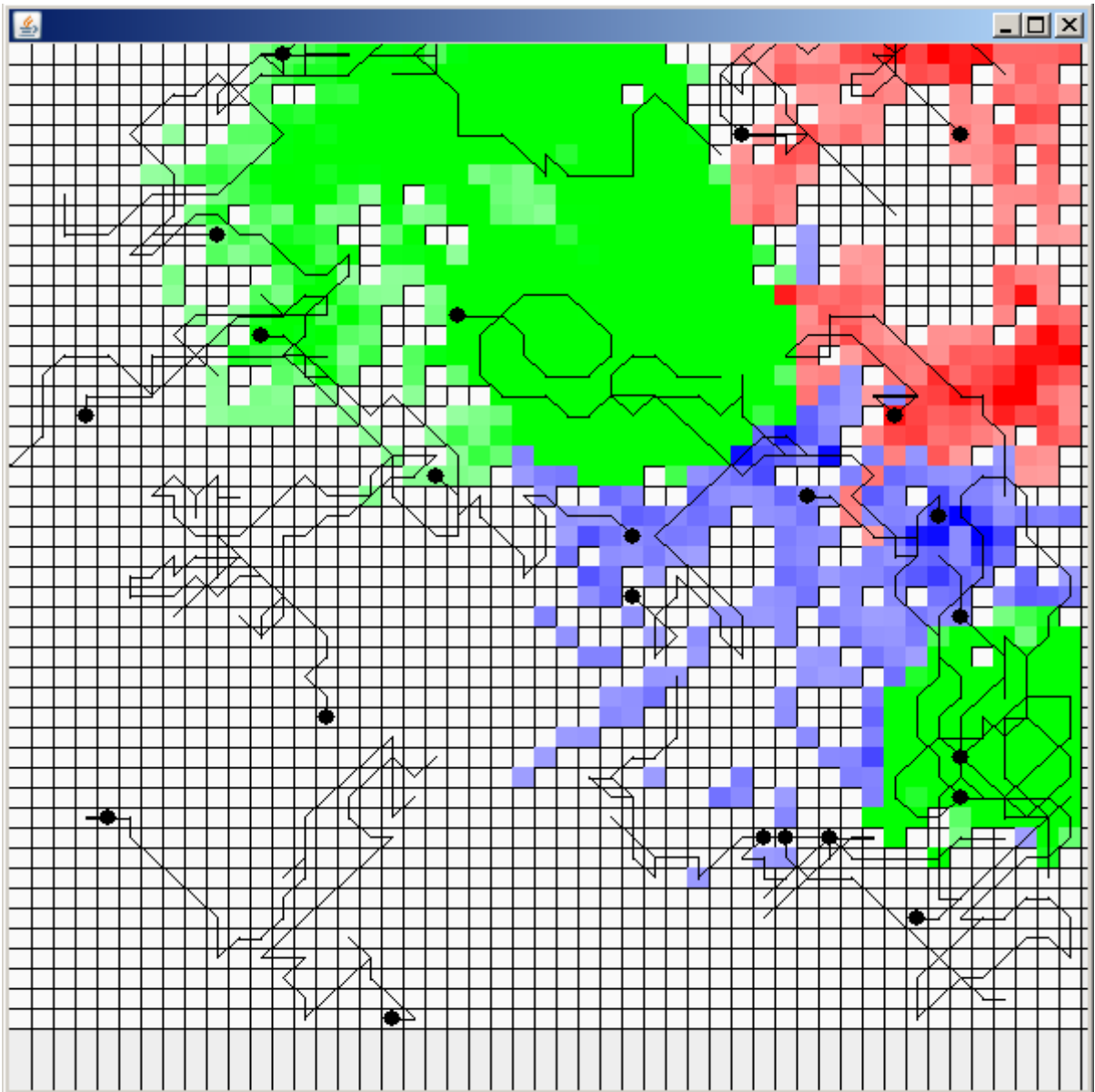
à 200 000 itérations, c'est presque fini



Le tas rouge en bas à gauche est très influent pour sa taille, il va bientôt disparaître.



Il aura donc fallu moins de 400 000 itérations pour arriver à ce résultat

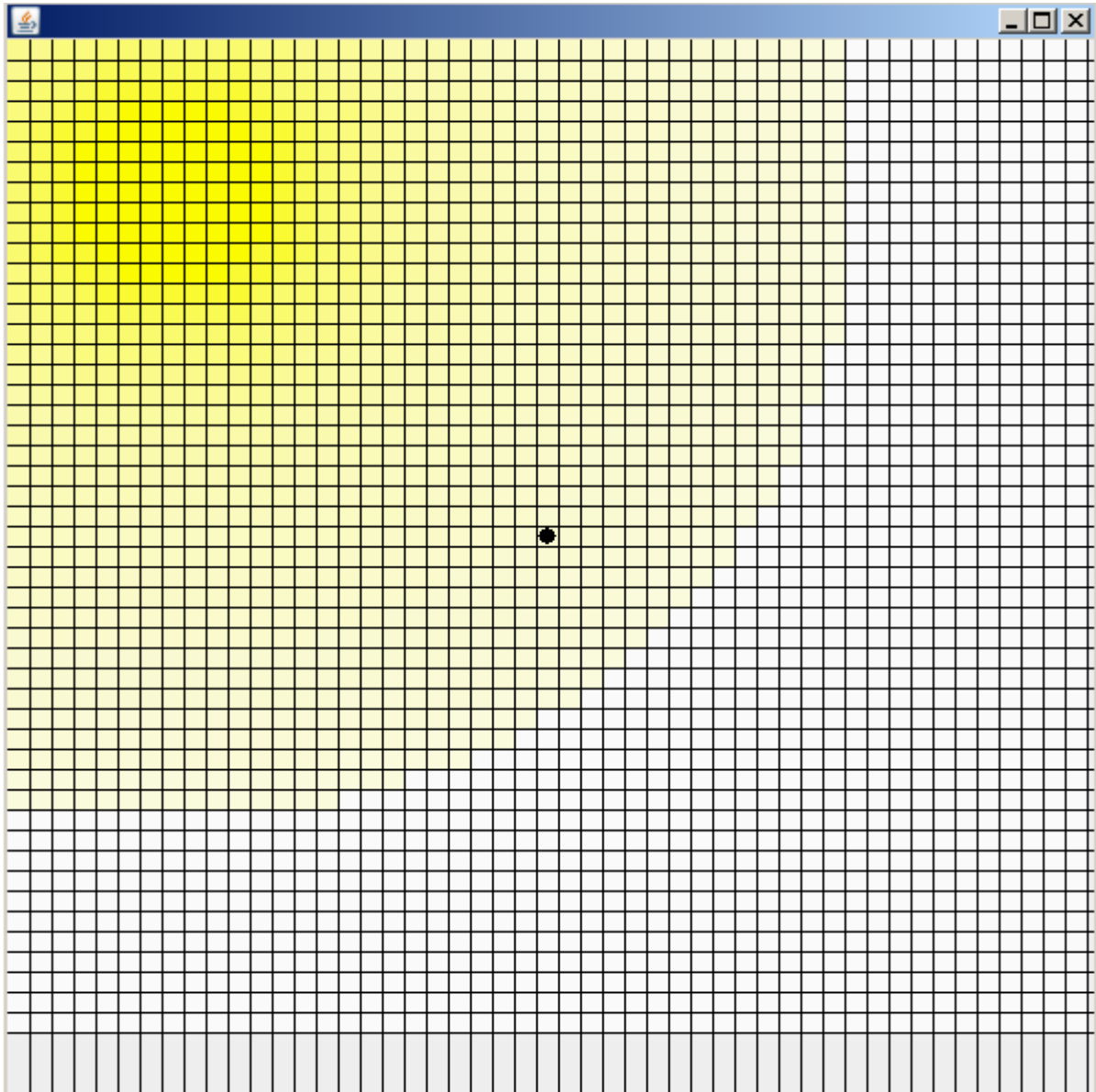


la carte associée au résultat.

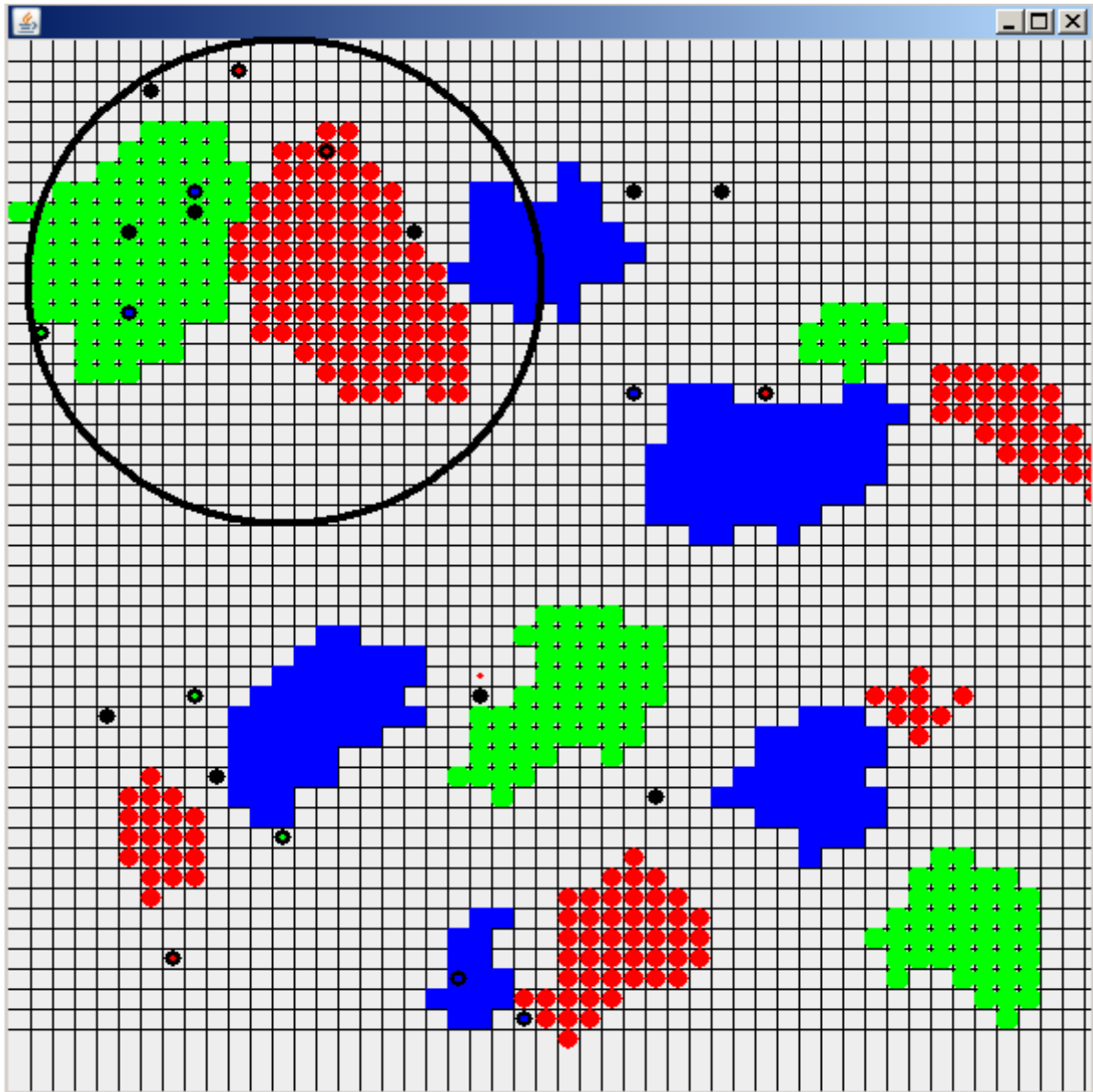
IV Nouvelles fonctionnalités

Le troisième système s'est vu ajouter trois nouvelles fonctionnalités : le point d'éclairage, l'histogramme et l'interface graphique.

Le point lumineux utilise le douzième élément des tableaux de la carte. La fonction blackHole associe à chaque case de la carte un poids qui dépend de la distance au centre, de la force et du rayon d'action du point lumineux. L'utilisation de la distance permet d'avoir soit un point très concentré de porté limité, soit un point de faible intensité mais s'étalant sur toute la surface..



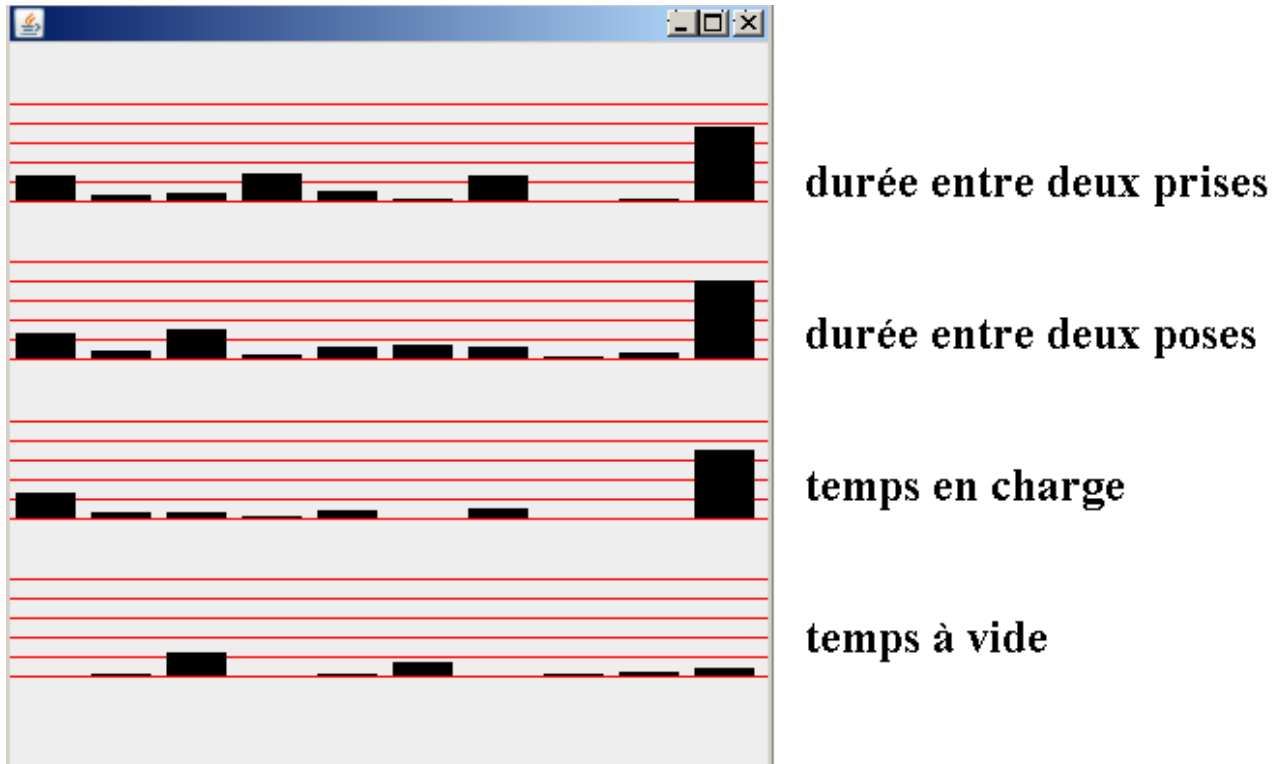
Le point lumineux, représenté sur la carte.



Dès le début, il va attirer les agents, et donc les tas vers lui. Si la puissance est trop grande, les agents resteront piégés.

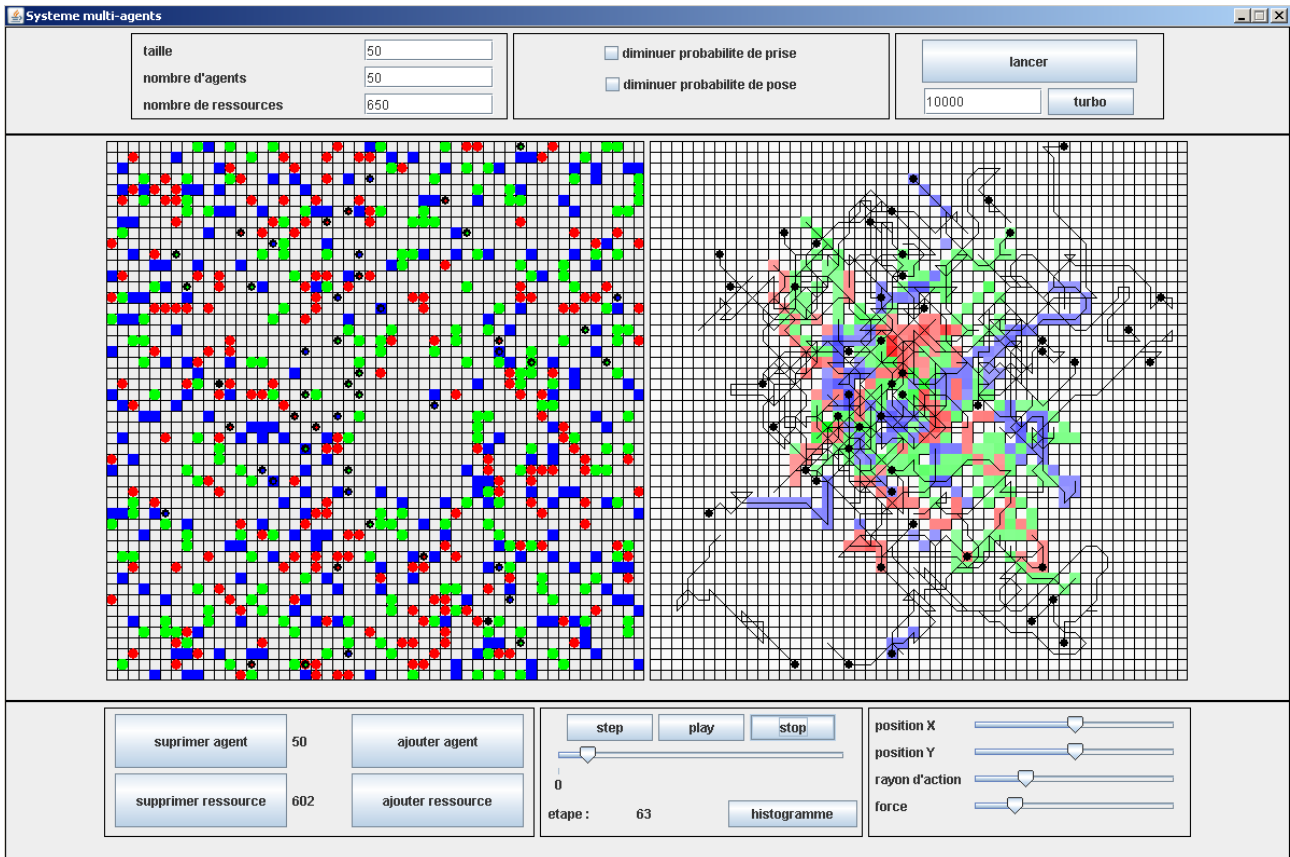
L'histogramme : il mémorise le nombre d'itérations qui séparent deux prises, deux poses, le nombre d'itération pendant laquelle l'agent a porté la même ressource et le temps passé à vide des mille derniers évènements de l'agent auquel l'histogramme est rattaché.

L'affichage graphique montre sous forme de barre ces quatres durées. L'échelle est un trait pour 100 itérations.



On remarquera que le temps à vide est souvent très faible : en effet, grâce aux disques d'influence, les agents trouvent rapidement une ressource à prendre.

L'interface permet d'observer le système multi-agent et d'interagir avec lui de manière intuitive. Voici les composants de l'interface :

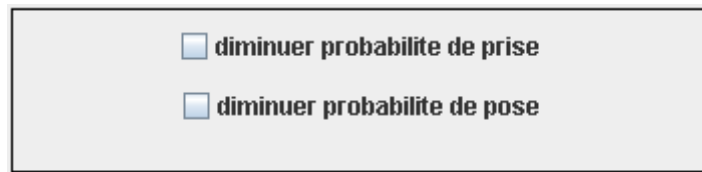


partie superieure :

Le panel d'initialisation : c'est ici que l'on donne les paramètres du système, c'est à dire la taille de la grille, le nombre d'agents et le nombre de ressources.

taille	<input type="text" value="50"/>
nombre d'agents	<input type="text" value="50"/>
nombre de ressources	<input type="text" value="650"/>

Le panel des probabilités : on peut modifier à tout moment la probabilité de prendre ou poser une ressource. Il faut cependant s'assurer, pour la pose de ressource, que la grille comporte au moins un tas d'au minimum deux éléments pour chaque ressources, sans quoi aucun tas ne se formera.

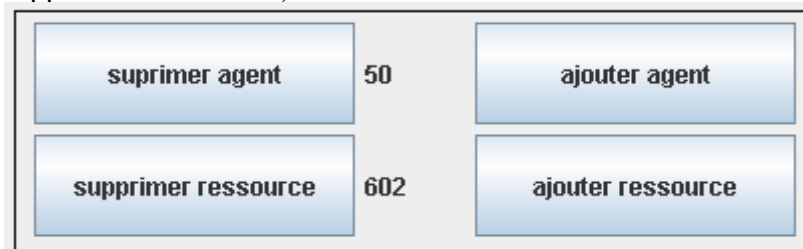


Le panel de lancement : permet d'initialiser un système à l'aide des paramètres du panel d'initialisation. Le bouton turbo permet de lancer le nombre de cycles spécifié à grande vitesse (il n'y a cependant pas de mise à jour de l'écran ni la possibilité d'utiliser l'interface pendant l'exécution).



Partie inférieure :

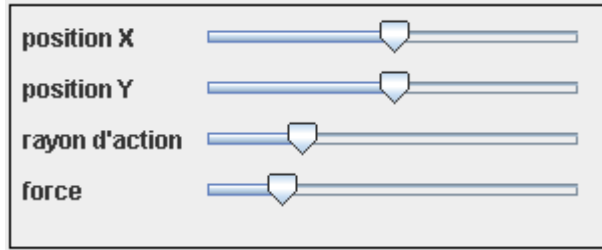
Le panel de gestion : affiche le nombre d'agents et de ressources sur la grille en temps réel. Permet l'ajout ou la suppression des agents et des ressources. (attention, si on supprime un agent transportant une ressource, on supprime la ressource).



Le panel de contrôle : permet de contrôler le déroulement des événements, avance, pause, pas à pas et réglage de la vitesse (en réalité la temporisation entre deux itérations).



Le panel du LightPoint : permet de gérer les paramètres du point lumineux : position, rayon d'action et puissance.



L'écran central : affiche l'état du système, à gauche la grille avec les agents et les ressources, à droite la carte avec les agents et leurs trente derniers déplacements, et les phéromones et le LightPoint.