

SMA - TP2 - Partie 1

Code et implémentation

Nous avons découpé notre projet de la façon suivante : Environnement gère la grille, Agent le comportement des agents, Runner lance la simulation et les classes Main lancent soit l'interface graphique soit l'interface ligne de commande.

A chaque étape de la simulation, on appelle la méthode "run" de chaque agent qui met à jour sa mémoire puis réalise les deux opérations de l'énoncé : "move" et "action".

Nous avons strictement suivi l'énoncé pour l'implémentation de ces deux méthodes, c'est-à-dire que l'agent se déplace dans une direction aléatoire et choisit une action suivant le conditionnement donné (voir le code pour l'implémentation, classe Agent).

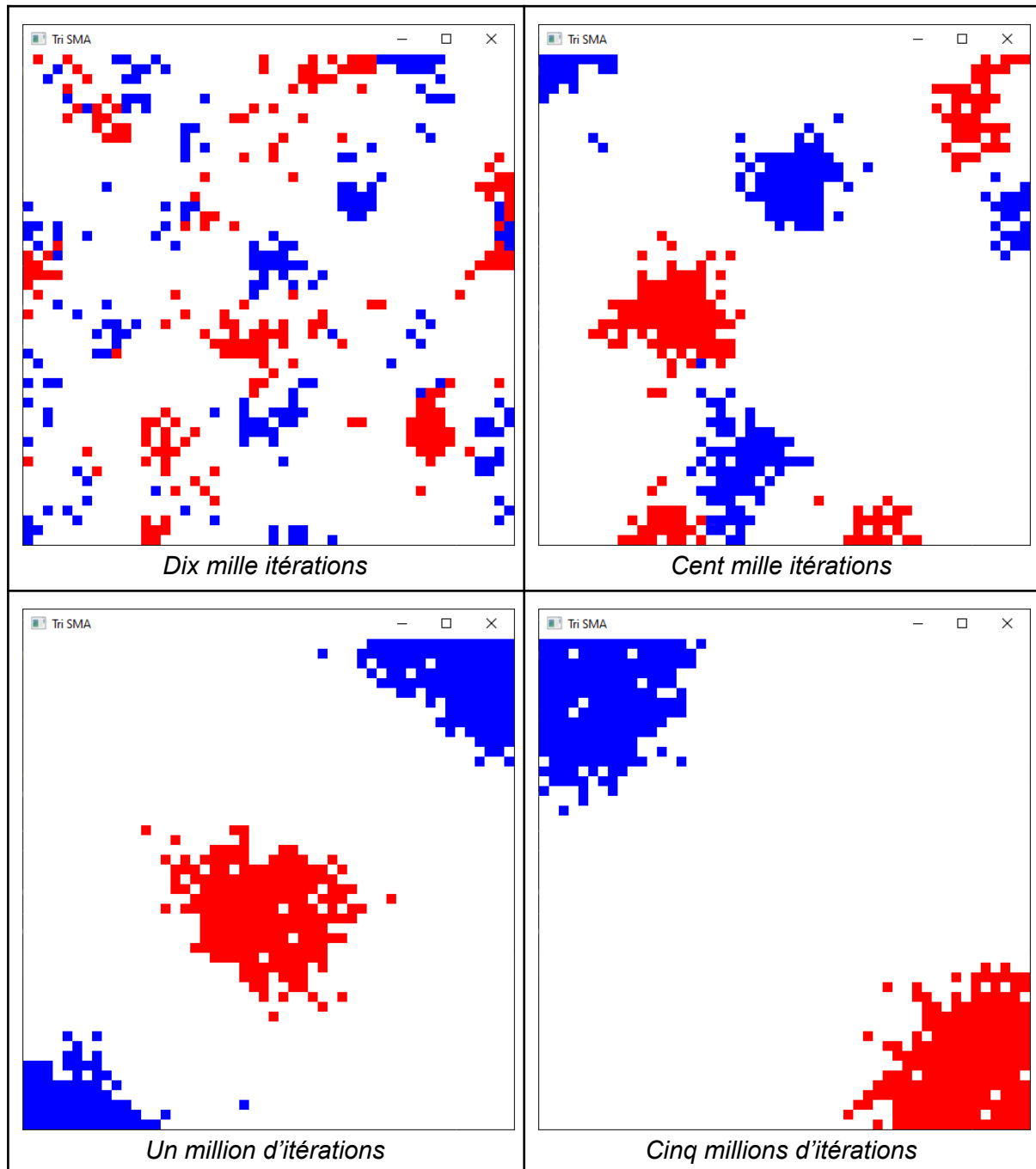
Pour l'application graphique, nous avons utilisé un pattern Observer/Observable classique avec le Runner lancé dans un thread séparé pour éviter que l'application graphique ne freeze. On met à jour l'affichage tous les 30 pas de temps pour éviter une surcharge de celui-ci.

Résultats - Sans erreur

Les agents ne faisant pas d'erreurs dans la discrimination entre les deux types d'objets A & B, on retrouve des résultats relativement cohérents avec la situation.

Plus les agents ont de temps pour se déplacer sur la grille et arranger les tas d'objets, plus ces derniers seront concentrés, gros et moins le nombre d'objets de l'autre type sera présent dans un tas qui ne lui correspond pas.

Note: les captures d'écran sont issues de simulations différentes.



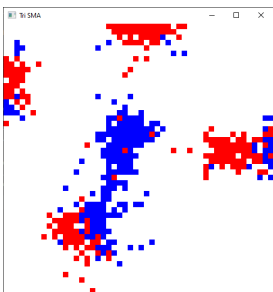
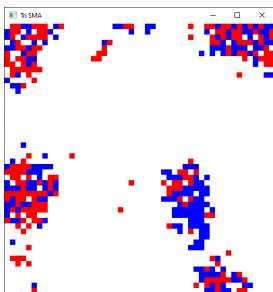
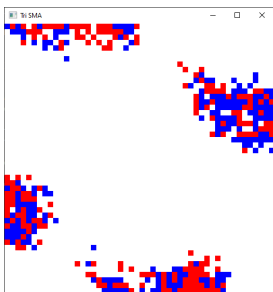
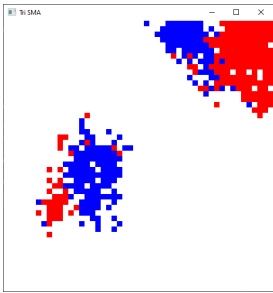
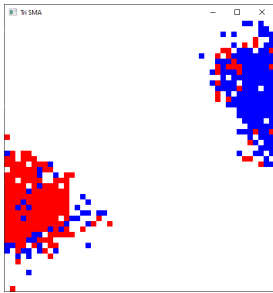
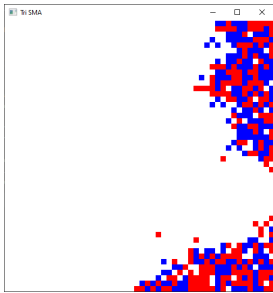
Résultats - Avec Erreurs

Ici, on ajoute comme demandé une erreur dans la capacité aux agents de distinguer les deux types d'objets.

Le comportement des agents est cependant pratiquement le même : ils vont chercher à regrouper les objets entre eux et à faire des clusters, à la simple différence qu'ils vont parfois se tromper et mettre des objets A avec des objets B, et vice-versa.

A noter que qu'au-delà de 0.5 pour le taux d'erreur, les résultats sont très semblables c'est pourquoi nous ne les avons pas inclus dans le tableau ci-dessous.

Note: les captures d'écran sont issues de simulations différentes.

| | 0.1 | 0.25 | 0.4 |
|------------|---|--|---|
| Cent mille |  |  |  |
| Un million |  |  |  |

Conclusion

Comme on peut l'observer, plus le nombre d'actions est élevé, plus le tri sera correct : il y aura plus de gros clusters et moins d'objets isolés.

L'influence du taux d'erreur est assez important puisque l'on remarque que l'on se retrouve très vite avec des clusters d'objets hétérogènes.

Ce taux va aussi inciter à créer plus rapidement un nombre réduit de gros clusters dans la mesure où les agents ont du mal à distinguer si c'est un "bon" cluster pour prendre/déposer ou non.

Cependant, si l'on augmente beaucoup le nombre d'actions, ces clusters seront potentiellement de plus en plus homogènes, à la condition que le taux d'erreur ne soit pas trop élevé.

Lien sans erreur (1 million) : <https://youtu.be/eNt5JJooiZM>

Lien avec erreur (1 million, taux 0.1) : <https://youtu.be/psG4FwJChXA>