

Systeme multi-agents

Hamdi Yacine p1709958 - Pont Louis p1704091

Lien vers le git : [SMA_TP2_Github](#)

Ce document résume le développement effectué lors du TP2.

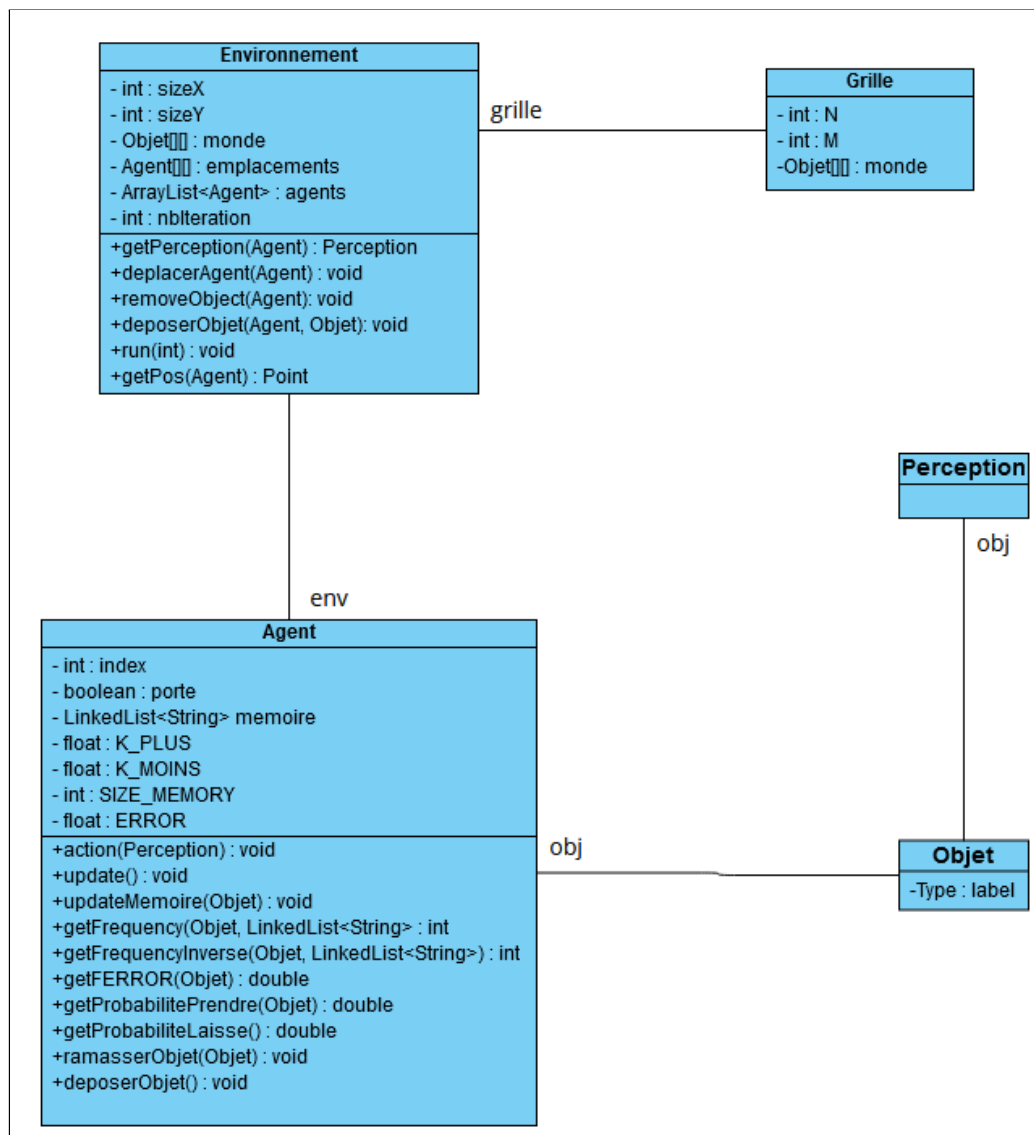


Diagramme de classe de l'application

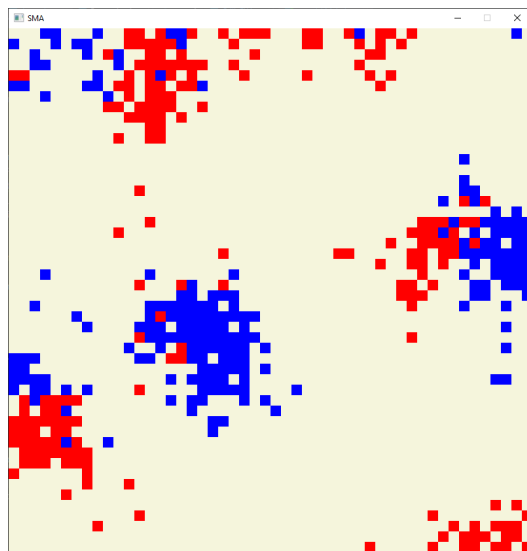
Dans cette première version, l'environnement s'occupe de déplacer les agents de manière aléatoire. A chaque itération, un agent récupère sa perception locale auprès de l'environnement, qui indique si l'agent se trouve sur un objet ou pas. L'agent réalise ensuite une action en fonction de cette perception :

- Si l'agent porte déjà un objet et qu'il n'y en a pas sur le sol, il va peut déposer l'objet
- Si l'agent ne porte pas d'objet et qu'il y en a un sur le sol, il va récupérer l'objet
- Dans les autres cas, il va se déplacer aléatoirement sur une des cases voisines.
- Il va ensuite mettre à jour sa mémoire

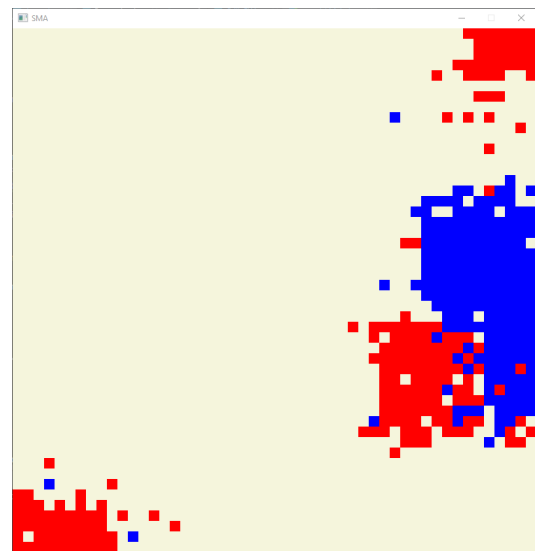
Le dépôt et la récupération d'un objet ne sont pas systématiques mais se font en suivant les formules de probabilité énoncées dans le sujet.

Le comportement principal que l'on remarque est la création de clusters : les agents regroupent les objets par type (grâce à leur capacité à les différencier). Leur performance à séparer les objets dépend du nombre d'itération, du taux d'erreur et des constantes de simulation :

- **Nombre d'itération** : plus le programme tourne et plus on converge vers une disposition en clusters stables.

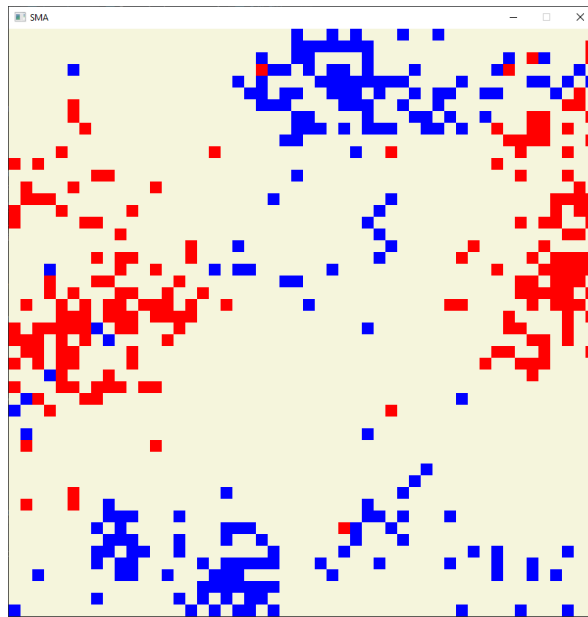


100_000 itérations



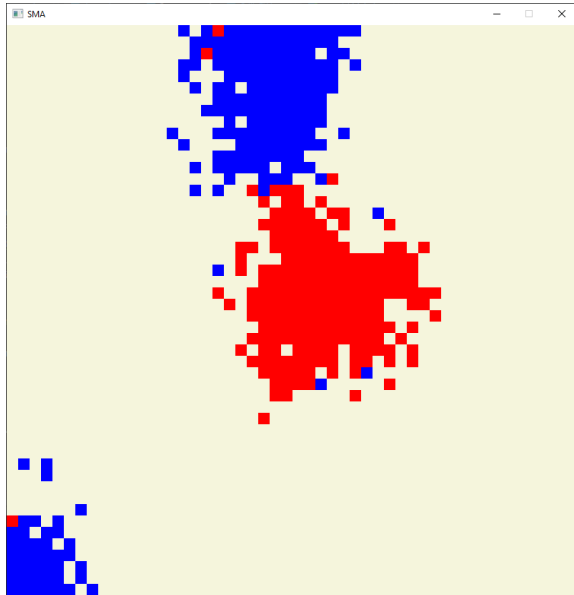
1_000_000 itérations

- **Constantes de simulation** : si K^+ est grand, les clusters ne sont pas stables car les agents ramassent les objets trop souvent. Si K^- est grand, la formation de clusters est plus efficace car les agents doivent rencontrer plus d'objets du même type avant de les déposer (ils s'assurent ainsi de déposer leur objet auprès d'un cluster pertinent). Plus la taille de la mémoire est grande, moins la création de clusters est efficace : les agents auront en effet moins tendance à prendre ou déposer un objet (probabilités trop faibles).

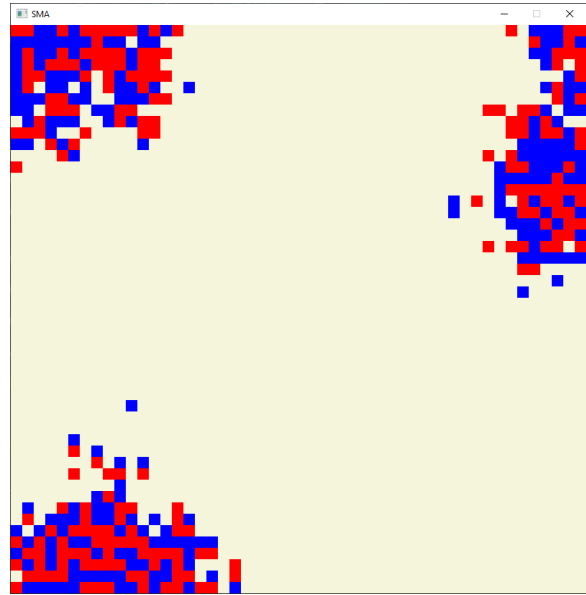


exemple avec $k^+ = 0.9$

- **Taux d'erreur** : plus le taux est grand, plus les objets sont mal placés. Avec un taux de 0.1, seulement quelques objets sont dans un mauvais groupe. Avec un taux d'erreur de 1, les agents forment des clusters indépendamment du type d'objet.



taux d'erreur de 0.1



taux d'erreur de 1

Pour gérer l'affichage graphique, nous avons utilisé JavaFX et le design pattern Observable/Observateur. L'interface graphique observe l'environnement, qui le notifie toutes les 100 itérations pour mettre à jour l'affichage. Une vidéo d'exemple est présente sur Github.