

5A INFO

TP

Tri collectif Multi-agents

ARI Moussa

LEFEBVRE Guillaume

Table des matières

[Introduction 3](#_Toc33393622)

[I. Évolution du projet 3](#_Toc33393623)

[II. Procédure de lancement 3](#_Toc33393624)

[III. Choix d’implémentations 5](#_Toc33393625)

[IV. Apport de chaque binôme 6](#_Toc33393626)

[V. Conclusion 6](#_Toc33393627)

# Introduction

Dans le cadre du TP de SMA, nous avons développé un système de tri collectif multi-agent dont le rôle est de trier de deux types objets A ou B dans un environnement. L’environnement est une grille sur laquelle on dispose initialement aléatoirement, des objets identifies par des lettres A ou B en quantités respectives *nA*, *nB* et un nombre *nbAgents* d’agents aussi répartis aléatoirement.

# Évolution du projet

Dans la première version du projet nous avons implémentées les principales classes notamment Agent, Grille, Coordonnées et avons implémentées le processus de perception et d’action des agents. Nous avons également implémenté le tri collectif mais avec un affichage dans la console du programme.

Dans cette version finale nous avons réalisé une interface graphique afin d’apercevoir plus facilement le déplacement des agents ainsi que des objets transportés.

# Procédure de lancement

Afin de lancer le programme il faut exécuter la classe Main.java dans laquelle sont déclarés les différents paramètres de la grille à savoir le nombre d’agents, d’objets, la taille de la mémoire, la valeur des constantes …etc.

Une fois exécuté, une fenêtre graphique s’affiche et invite l’utilisateur à cliquer sur le bouton *start* chargé de lancer la simulation du tri collectif.

Une image contenant appareil, climatiseur

Description générée automatiquement

Figure : Initialisation Grille 30\*30

Une image contenant assis, suivant, vert, ciel

Description générée automatiquement

Figure : Tri collectif en cours

Nous constatons sur la fig. 2, les agents représentés par les blocs verts entrain de déplacer les objets.

Représente un agent libre

Représente un objet de type A

Représente un objet de type B

Représente un agent transportant objet de type A

Représente un agent transportant un objet de type B

# Choix d’implémentations

Une image contenant capture d’écran

Description générée automatiquement

Nous avons développé au total 8 classes afin de rendre modulaire et réutilisable le code. Nous avons utilisé aussi la classe Thread de JAVA et le principe de verrou avec le mot clé *synchronized* afin de respecter le parallélisme.

# Apport de chaque binôme

Le travail au sein de notre binôme a été reparti équitablement et de manière collaborative. Nous faisons les réflexions ensembles en proposant chacun son idée et en sélectionnant celle qui nous semble la plus pertinente. Afin d’implémenter le programme, nous avons identifié́ ensemble les différentes fonctionnalités. Ainsi nous avons pu déléguer les taches de manière complémentaire mais aussi de façon de collaborative pour certaines tâches.

# Conclusion

Ce Projet nous initié à la programmation de système multi agent à travers un cas concret. Nous l’avons trouvé enrichissant car il nous a permis de comprendre les principes théoriques abordés en cours et d’acquérir des compétences pratiques notamment l’utilisation de verrou sur les méthodes pour les opérations indivisibles dans le langage JAVA.