



Projet Java: Gestion des Robots

15/05/2025

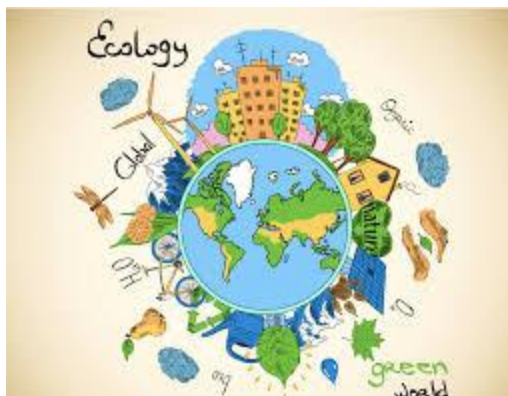
Prof. Dkhili Amira

Binome:

Belabed Youssef RT 2-1

Achouri Mohamed RT 2-3

INSAT 2024/2025



Introduction

Dans un contexte mondial marqué par la transition énergétique et la nécessité de préserver l'environnement, l'automatisation des tâches à faible impact écologique devient une priorité. Ce projet a pour objectif de développer une application Java permettant de gérer une flotte de **robots écologiques** chargés de réaliser diverses missions respectueuses de l'environnement, telles que la collecte de déchets, la surveillance d'espaces naturels, ou encore l'entretien de zones vertes.

L'application offre une interface de gestion centralisée qui permet de suivre en temps réel l'état, la localisation et les tâches de chaque robot. Elle assure également une coordination intelligente entre les robots, en optimisant leurs déplacements et leurs interventions pour minimiser la consommation énergétique. Ce projet s'inscrit dans une démarche à la fois technologique et durable, illustrant l'apport de l'informatique dans la construction d'un avenir plus vert.



Objectifs

1. Développer une application Java de gestion de robots en suivant les principes de la POO (polymorphisme, héritage, gestion des exceptions, interface, interface graphique, etc)
2. Intégrer des considérations écologiques dans la conception

Specifications

Le système développé vise à simuler et gérer le comportement d'un **robot de livraison écologique** optimisé pour la consommation énergétique. Ci-dessous les spécifications fonctionnelles et techniques du projet :

1. Objectif principal

Concevoir une application Java permettant de gérer un robot de livraison autonome qui choisit systématiquement **le chemin le plus court** entre un point de départ et un point de livraison, afin de **minimiser la consommation d'énergie**.

2. Fonctionnalités principales

- **Modélisation de l'environnement** : représentation de l'espace sous forme de grille ou de graphe (carte avec des points et des chemins).
- **Définition des points de livraison** : possibilité de définir dynamiquement des points de départ et d'arrivée.
- **Calcul du chemin optimal** : utilisation d'un algorithme de plus court chemin (comme Bellman) pour déterminer le trajet le plus économe en énergie.
- **Suivi du robot** : affichage de la position actuelle du robot et du chemin parcouru.
- **Simulation des livraisons** : possibilité de lancer des missions de livraison successives avec recalcul automatique du trajet optimal.

3. Contraintes écologiques

- Le robot doit **éviter les zones à forte consommation d'énergie** (pentes, zones à trafic élevé, etc.).
- Il privilégie les trajets courts et plats pour réduire l'utilisation de sa batterie.
- Possibilité d'intégrer une estimation de **consommation énergétique par trajet**.

4. Technologies utilisées

- Langage de programmation : **Java**
- Structures de données : **graphe pondéré, file de priorité**
- Algorithmes : **Dijkstra** ou **A*** ou Bellmann pour le calcul du plus court chemin
- Interface (optionnelle) : console ou interface graphique simple pour visualiser le robot et son environnement

5. Évolutivité

- Ajout possible de plusieurs robots livrant en parallèle

- Gestion de recharge automatique du robot en cas de batterie faible

Choix techniques:

On a implemente 3 classes:

Robot et RobotConnecte → abstraites

RobotLivraison → Concrete

Pour l'Heritage:

RobotLivraison Herite de RobotConnecte ui lui meme herite de Robot

Pour le Polymorphisme:

Il ya des méthodes abstraites à concrétiser ainsi que des méthodes à redéfinir

Pour les interfaces:

On a cree un interface Connectable implemente par RobotConnecte

Pour l'interface Graphique:

On va créer un Control Panel qui va gérer le robot ainsi qu' un modèle qui représente le mouvement du robot

Pour la gestion des exceptions:

On a implementer les classe EnergieInsuffisanteException et MaintenanceRequiseException aui extend RobotException

Choix Ecologiques:

Notre choix écologique est l'économie de l'énergie en suivant le chemin le plus court.

Pour pouvoir implémenter ceci on va s'inspirer d'un des tp de ce semestre le TP1

En fait en vas modéliser les destinations par une réseau des stations et on va utiliser l'algorithme de Bellman pour calculer le chemin le plus court

Le deuxième choix écologique est le fait qu'un robot de livraison émet moin de CO2 que les moyens de transport utilisés ce qui garantit que notre projet est écologique et vert (pas de carbon footprint)



Conclusion

Ce projet de robot de livraison écologique s'inscrit dans une démarche innovante alliant technologie et développement durable. Grâce à l'utilisation d'algorithmes d'optimisation tels que Bellman, le robot est capable de choisir le chemin le plus court, réduisant ainsi sa consommation énergétique tout en assurant l'efficacité des livraisons.

L'approche modulaire et évolutive de l'application permet d'envisager de nombreuses améliorations, comme la gestion de plusieurs robots, l'adaptation à des environnements dynamiques ou encore l'intégration de critères écologiques plus complexes. Ce projet constitue ainsi une base solide pour développer des solutions robotiques intelligentes et respectueuses de l'environnement.

