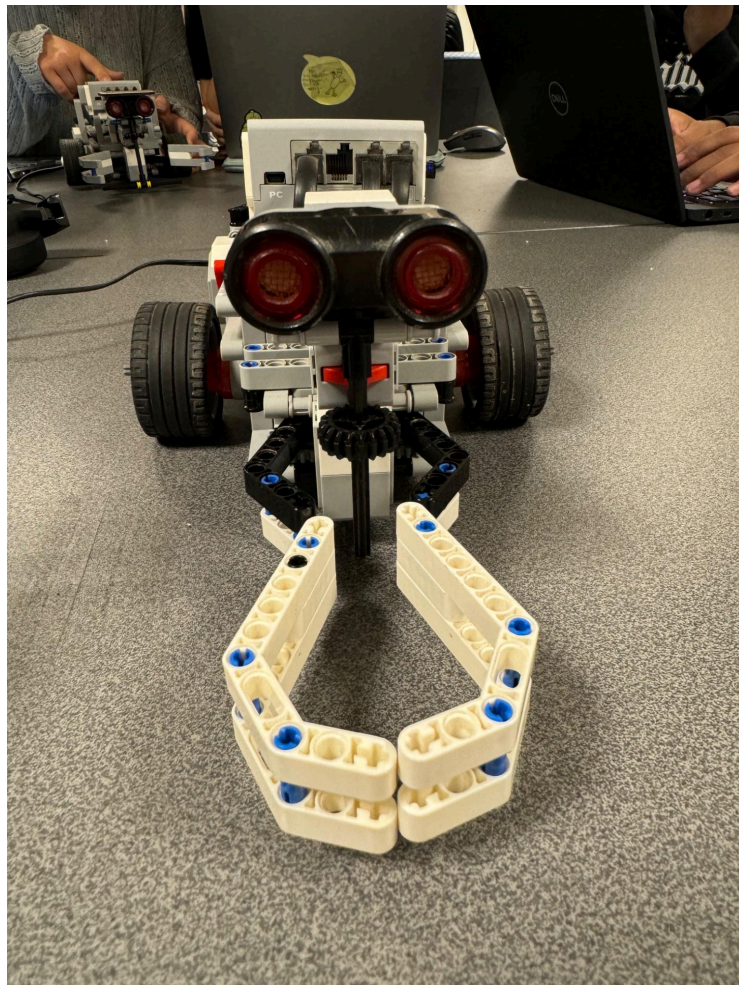


# Rapport Final



Mathias Devilliers n°12201983

Carole Mitton n°11906540

Paul Ndong n°12120816

Saliha Ozturk n°12012637

# Table des matières

<b>1.Introduction.....</b>	<b>3</b>
<b>2.Stratégies adoptées.....</b>	<b>3</b>
2.1. Au niveau de la programmation du robot.....	3
2.1.1. La position du robot.....	3
2.1.2. Évitements des objets.....	3
2.2 Aux niveau du travail collectif.....	4
2.2.1. Équipe théorique vs équipe pratique.....	4
2.2.2. Réunions d'équipe à chaque début et chaque fin de cours.....	4
2.2.3. Travail en dehors des heures de cours, dans le FabLab.....	4
<b>3. Forces et faiblesses du groupe.....</b>	<b>4</b>
3.1. Forces.....	4
3.2. Faiblesses.....	5
<b>4. Piste d'amélioration.....</b>	<b>5</b>
<b>5. Guide de lecture.....</b>	<b>6</b>
<b>6. Glossaire.....</b>	<b>6</b>
<b>8. Index.....</b>	<b>7</b>

## 1.Introduction

Ce document a pour objectif d'analyser et de discuter des stratégies adoptées au cours du projet, en mettant en lumière les forces et les faiblesses rencontrées par le groupe. Cette analyse vise à identifier les éléments à reproduire ou à éviter lors de futurs projets

## 2.Stratégies adoptées

Pour garantir la réussite du projet et optimiser la collaboration au sein de l'équipe, plusieurs stratégies ont été adoptées tout au long de la phase de développement du robot LEGO EV3.

### 2.1. Au niveau de la programmation du robot

#### 2.1.1. La position du robot

Nous avons choisi pour notre robot de lui implémenter des paramètres qui lui permettent de connaître sa position dans l'espace avec des coordonnées cartésiennes en x et y ainsi que son orientation.

Toutes ces informations étaient remises à jour à chacun de ses déplacements grâce à des méthodes que nous avons codées pour calculer ses nouvelles coordonnées en fonction des distances parcourues. Cependant, les mouvements du robot ne sont jamais exactement ce que l'on attendait. Par conséquent, il faut pouvoir mettre à jour les paramètres de positions avec des informations que l'on peut trouver sur le terrain et qui sont donc toujours les mêmes. Par exemple, arrivé à la ligne blanche, le robot va regarder en face de lui sur un petit angle pour trouver la distance la plus courte entre lui et le mur et ainsi se réorienter bien en face par rapport au mur.

Avoir accès aux coordonnées du robot nous a permis de coder différentes méthodes permettant au robot de retourner au centre, de savoir si un palet qu'il a vue était en réalité dans un camp ou s'il était sur le terrain ou encore sur quel angle il devrait faire une recherche pour avoir le plus de chance de voir des palets.

Par contre, pour que cela fonctionne, il est essentiel que le robot ne se retrouve jamais à rouler dans un mur ou un obstacle sinon l'intégralité des coordonnées sont alors faussées et il devient impossible de prévoir ce que le robot pourrait faire.

#### 2.1.2. Évitements des objets

Dans les différentes méthodes pour faire avancer notre robot, nous avons ajouté des conditions s'il rencontre un objet en vue à moins de 20 centimètres. Nous avons estimé qu'il y avait deux cas possibles. Soit l'objet devant nous ne bouge pas du tout et alors nous devons nous déplacer, soit l'objet va bouger et nous pourrions reprendre notre route, si c'est un robot qui passe devant nous par exemple. C'est pour cela que notre robot, lors de ses méthodes pour avancer, s'il détecte quelque chose à moins de 20 centimètres, alors il va s'arrêter quelques secondes puis regarder à nouveau. S'il n'y a plus rien, alors il reprend son chemin. S'il y a encore quelque chose, il va se déplacer vers un autre endroit. Grâce à ces méthodes nous pouvons assurer que notre robot n'avancera pas dans le vide contre un mur ou contre un autre robot, évitant ainsi de fausser toutes les coordonnées. Pour autant, nous ne sommes pas à l'abri qu'un robot nous percute.

## 2.2 Aux niveau du travail collectif

### 2.2.1. Équipe théorique vs équipe pratique

Afin de mieux structurer le travail, nous avons choisi de diviser l'équipe en deux groupes distincts : un groupe théorique et un groupe pratique. Le groupe théorique est chargé de concevoir l'algorithme de commande du robot, de définir les stratégies de collecte des palets et de résoudre les problèmes théoriques liés à la planification et à la prise de décision. Le groupe pratique, quant à lui, se concentre sur l'implémentation du code sur le robot et sur les tests pratiques dans le cadre de la compétition. Cette division permet de renforcer l'efficacité de chacun en fonction de ses compétences spécifiques, tout en assurant que les deux aspects (théorique et pratique) se complètent.

### 2.2.2. Réunions d'équipe à chaque début et chaque fin de cours

Nous avons mis en place une routine de réunions d'équipe systématiques. Ces réunions ont lieu à chaque début et à la fin de chaque cours. En début de séance, nous discutons des objectifs du jour, des tâches à accomplir et des obstacles éventuels rencontrés. À la fin, nous faisons un point sur les avancées réalisées, les ajustements nécessaires et les prochaines étapes à suivre. Ces réunions permettent de maintenir une communication fluide et de s'assurer que tous les membres de l'équipe sont sur la même longueur d'onde tout au long du projet.

### 2.2.3. Travail en dehors des heures de cours, dans le FabLab

Le groupe a également pris l'initiative de travailler en dehors des heures de cours, en particulier dans le FabLab de l'université.

## 3. Forces et faiblesses du groupe

### 3.1. Forces

L'une des principales forces de notre équipe réside dans la communication fluide et régulière entre ses membres. Grâce à cette communication, nous avons pu identifier rapidement les problèmes et trouver des solutions ensemble, évitant ainsi des malentendus ou des erreurs qui auraient pu compromettre notre progression.

De plus, l'entraide a permis une bonne entente des membres du groupe tout au long du projet. Dès qu'un membre rencontrait des difficultés, les autres étaient là pour apporter leur aide. Cette dynamique d'entraide a renforcé notre cohésion d'équipe et a permis à chacun de progresser, même lorsqu'il était confronté à des obstacles.

Grâce à la répartition des tâches, chacun a pu se concentrer sur des tâches qui correspondaient à ses forces, tout en restant impliqué dans les autres aspects du projet. Cette approche a permis une avancée efficace et a réduit le risque de surcharge de travail pour un membre en particulier. De plus, la clarté dans la répartition des tâches évite les conflits ou les zones de flou concernant les responsabilités de chacun.

Au début et à la fin de chaque cours, des réunions régulières ont été organisées. Ces réunions ont permis de garder une vue d'ensemble sur l'avancement.

Enfin, l'une des forces majeures a été notre capacité à travailler de manière régulière. Le groupe a respecté les engagements et s'est assuré de faire des progrès constants, même en dehors des heures de cours.

### 3.2. Faiblesses

L'une des faiblesses du groupe a été la tendance à vouloir perfectionner le travail. Bien que l'attention aux détails soit importante, cela nous a fait perdre du temps sur des éléments qui n'avaient pas un impact significatif sur la performance ou la réussite du projet.

De plus, une des faiblesses du groupe a été l'hésitation à solliciter de l'aide auprès du professeur. Bien que nous ayons eu quelques questions ou doutes, nous avons souvent préféré chercher des solutions par nous-mêmes, parfois au détriment de notre efficacité. Une demande d'aide plus fréquente ou une consultation plus régulière avec notre professeur aurait pu nous permettre de gagner du temps et d'éviter certains obstacles techniques ou théoriques.

## 4. Piste d'amélioration

Dès le début du projet nous aurions dû établir des normes de codage à respecter. Pour les prochaines fois nous définirons à l'avance les différentes normes que nous devons suivre sur la manière dont nous nommons nos méthodes ou nos classes par exemple. En choisissant la langue dans laquelle elles seront toutes nommées, la forme qu'elles doivent prendre. Pour rendre la navigation plus facile dans le code.

Aussi, l'utilisation de github est très intéressante pour les travaux de groupe de ce genre mais notre compréhension de son fonctionnement nous a rendu réticent à l'utiliser et les erreurs que l'on faisait en nous en servant nous prenaient souvent un peu de temps à corriger. En prenant le temps d'avoir une meilleure compréhension du fonctionnement de github nous pourrions travailler plus efficacement en mettant plus simplement et sans erreur notre travail en commun.

## 5. Guide de lecture

Ce document présente une analyse détaillée des stratégies adoptées, des forces et des faiblesses du groupe 1 dans le cadre d'un projet d'IA avec le robot LEGO EV3. Le lecteur peut utiliser les informations ci-dessous pour se repérer facilement dans le contenu et en saisir les principaux enjeux.

## 6. Glossaire

**Asynchrone** : Mode de fonctionnement où le robot exécute une tâche sans attendre qu'elle soit terminée avant de passer à une autre.

**FabLab** : Espace de travail collaboratif pour la réalisation de projets technologiques.

**Github** : Plateforme de gestion de versions permettant de collaborer sur des projets de programmation.

**Camp adverse** : Zone située en face de la position de départ du robot sur le terrain de jeu.

**Coordonnées cartésiennes** : Position d'un point sur un plan défini par deux axes (X et Y).

**LeJOS** : Framework Java utilisé pour programmer les robots LEGO EV3.

**Palet valide** : Palet détecté et récupérable selon les conditions définies dans le cahier des charges.

**Pince** : Partie mécanique du robot permettant de saisir un palet.

**Terrain quadrillé** : Aire de jeu où le robot évolue, marquée par des lignes de différentes couleurs servant de référence pour le positionnement.

## 7. Références

### 1. Documentation officielle LEGO EV3 et LeJOS

<https://lejos.sourceforge.io/ev3/docs/overview-summary.html>

### 2. Cahier des charges

<https://github.com/MathiaasGH/wall-e-dmno/blob/main/CDC%20folder/CDC.pdf>

### 3. Plan de développement

<https://github.com/MathiaasGH/wall-e-dmno/blob/main/CDC%20folder/PlanDeD%C3%A9veloppement.pdf>

#### 4. Plan de test

#### 5. Projet LEGO EV3 : Intelligence Artificielle

[https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:ia:project\\_lego](https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:ia:project_lego)

#### 6. Cours MIASHS - Intelligence Artificielle

<https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:robotics>

## 8. Index

**Camp adverse** : Section 2.1.1

**Collision** : Section 2.1.2

**Coordonnées cartésiennes** : Section 2.1.1

**FabLab** : Section 2.2.3

**Github** : Section 4.1

**LEGO EV3** : Section 1, Section 2

**Palet** : Section 2.1.1

**Pince** : Section 2.1.1

**Réunions d'équipe** : Section 2.2.2