

# Rapport

## Projet Intelligence Artificielle



PAUL Paco  
MEYRIEUX Gautier  
BERGÈRE Clement  
CHAPPOT Alexandre

L3 MIASHS  
Année 2023-2024

# **Sommaire :**

<b>Les Réussites</b>	<b>3</b>
Réussite N°1 : La détection des palais	3
Réussite N°2 : La position du robot	3
Réussite N°3 : L'arborescence	3
<b>Les Échecs</b>	<b>4</b>
Echec N°1 : Capteur couleur	4
Échec N°2: Gestion du temps	4
Échec N°3 : Utilisation de GitHub	4
<b>Améliorations possibles sur le projet</b>	<b>5</b>
Les axes d'amélioration généraux	5
Les axes d'amélioration sur le robot	5

# Les Réussites

## Réussite N°1 : La détection des palais

Malgré le fait que notre robot est imprécis et rate parfois des palets, notre programme de détection via le capteur ultrason est probablement ce qui nous a démarqué des autres groupes, nous permettant de ramener à coup sûr 3 palets, voir plus si nous avons assez de temps, ou un robot plus rapide, jusqu'à 5.

Pour choisir quel palet aller chercher, notre robot tourne en direction du palet le plus proche qu'il croit présent grâce à leurs coordonnées, vise le point le plus proche détecté  $\pm 15^\circ$  par rapport à la position supposé pour une meilleure chance de détection.

L'un des problèmes est qu'à cause du manque de temps, il ne sait pas en réalité si le point le plus proche est bien un palet, un robot ou un mur ce qui aurait pu se faire relativement simplement. Mais il valait mieux celà plutôt qu'une recherche encore plus aveugle.

## Réussite N°2 : La position du robot

Nous avons intégré à notre robot un système de de coordonnées qui est actualisé à chaque déplacement, permettant au robot de savoir où il se trouve et dans quelle direction s'orienter pour aller marquer. De plus, afin de minimiser les erreurs, à chaque palais marqué sa coordonnée X est remplacée par la coordonnées X de la ligne blanche. Néanmoins nous n'avons pas implémenté cela pour les Y car le capteur couleur n'était pas entièrement fonctionnel mais cela aurait permis de mettre à jour les coordonnées du robot à chaque passage sur une ligne de couleur. De plus l'orientation n'est pas vérifiée et corrigée si nécessaire lorsque le robot passe la ligne blanche en utilisant le mur face à lui pour limiter les erreurs dans la position.

## Réussite N°3 : L'arborescence

Les différentes classes créées et l'organisation utilisée lors de notre projet ont permis une avancée rapide, en organisant le tout selon les capteurs pour pouvoir travailler séparément.

Les capteurs étaient fonctionnels rapidement, ce qui a permis d'être efficace sur les dernières semaines quand le nombre de tests s'intensifiaient.

# Les Échecs

## Echec N°1 : Capteur couleur

Lors du développement de notre projet, nous avons rencontré des difficultés à implémenter une partie importante : le calibrage du capteur ColorSensor sur le plateau de jeu.

L'objectif initial étant de relever précisément les valeurs des différentes bandes colorées du plateau en utilisant une méthode qui déplace le robot sur ces bandes jusqu'à obtenir des intervalles de valeurs RGB satisfaisants.

Cependant, des idées d'utilisation de package ont posé des problèmes dans leur mise en oeuvre comme la librairie de Lejos NTXcolorsensor

Cette dernière offrait des fonctions prédéfinies pour chaque couleur du spectre RGB (rouge, vert, bleu) ex : getRedMode(), qui est censée retourner les valeurs de rouge lorsque le capteur est placé dessus. Malheureusement, aucun test n'a abouti.

## Échec N°2: Gestion du temps

Malgré notre réussite au concours de robot, la gestion du temps a été un défi majeur tout au long du projet. En raison de divers obstacles et contraintes, nous n'avons pas eu suffisamment de temps pour tester et implémenter des méthodes cruciales pour le bon fonctionnement du robot (notamment un suivi plus précis des coordonnées avec les lignes sur le plateau et donc le colorSensor, une vérification de l'orientation du robot, la vérification des palet qui aurait put être bien meilleur et précise, ...) ce qui nous à conduit à restreindre les capacités du robot.

## Échec N°3 : Utilisation de GitHub

Bien que nous ayons finalement réussi à utiliser GitHub plus ou moins convenablement, l'utilisation initiale de la plateforme a été difficile et périlleuse. La complexité de Git lorsque l'on ne l'a jamais utilisé et la gestion des conflits ont entraîné des retards et de la perte de code dans le développement. Une connaissance plus approfondie sur l'utilisation de GitHub aurait probablement amélioré notre efficacité et limiter la perte de temps.

## Améliorations possibles sur le projet

## Les axes d'amélioration généraux

- Avoir une meilleure connaissance dans l'utilisation de GitHub.
- Planifier et répartir les tâches plus tôt dans le lancement du projet.
- Faire des tests plus régulièrement et de manière plus efficace du code.
- Avoir une meilleure gestion du temps global .

## Les axes d'amélioration sur le robot

- Augmenter la vitesse de déplacement tout en gardant la précision de l'actualisation des coordonnées.
- Faire des déplacements sous forme de tangente au lieu d'avancer, tourner puis avancer pour éviter un obstacle.
- Piloter les pinces lorsque le robot est en mouvement afin de gagner du temps.
- Utiliser les lignes de couleur autres que la blanche pour mettre à jour les coordonnées dès qu'on passe dessus afin de minimiser les erreurs .
- Utilisation de la caméra au-dessus du plateau.