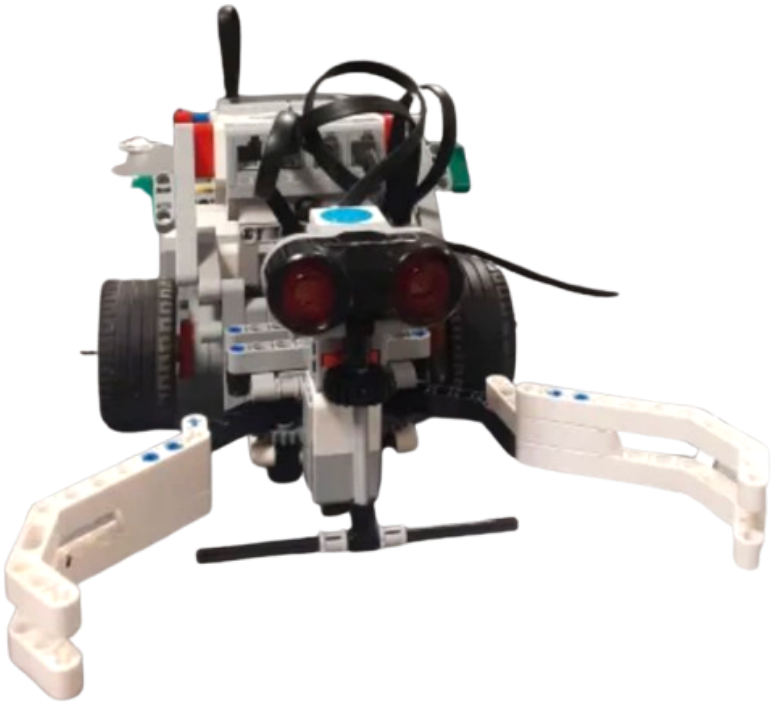
Plan de Tests

Robot Autonome EV3 Lego Mindstorm



CHERQAOUI Yassmina
LAGET-THOMAS Zoé
NEZIRAJ Narta
UNAL Basak

L3 MIASHS

2024/2025

Sommaire

1. Introduction

Ce document décrit le plan de tests conçu pour valider le fonctionnement du robot autonome développé dans le cadre de notre projet. Ce robot est chargé de naviguer de manière autonome, de collecter des palets, d'éviter des obstacles et de déposer les palets sur une ligne cible. L'objectif principal du plan de tests est de garantir que toutes les fonctionnalités répondent aux exigences spécifiées et que le robot fonctionne de manière fiable dans un environnement complexe.

Pour atteindre cet objectif, le plan de tests s'appuie sur une méthodologie rigoureuse, incluant des tests unitaires, d'intégration et fonctionnels. Chaque type de test a été conçu pour couvrir les différents aspects du système, depuis les composants individuels jusqu'à leur interaction globale, tout en prenant en compte les scénarios réalistes auxquels le robot sera confronté. En outre, ce plan permet d'identifier rapidement les problèmes et d'évaluer l'efficacité des solutions apportées.

2. Approche du Plan de Tests

2.1. Types de Tests

- **Tests Unitaires :** Vérification des composants individuels (moteurs, capteurs, méthodes des classes).
- **Tests d'Intégration :** Validation des interactions entre les composants (moteurs, capteurs, pince).
- **Tests Fonctionnels :** Validation du comportement global du robot dans des scénarios réalistes (navigation, collecte/dépôt, évitement d'obstacles).

2.2. Stratégie de Tests

- Priorisation des scénarios critiques (navigation droite, collecte/dépôt, évitement d'obstacles).
- Test itératif : résolution des erreurs détectées avant de passer aux étapes suivantes.
- Test progressif : des tests unitaires vers les tests fonctionnels pour construire la robustesse progressivement.

2.3. Critères de Réussite

- Navigation droite avec un écart <5% de la trajectoire.
- Détection des obstacles >90% de fiabilité à une distance de 30 cm.
- Collecte et dépôt de palets sans erreur dans 90% des cas.
- Achèvement des tâches en moins de 2 minutes dans des scénarios réalistes.

3. Tests Unitaires

Classe	Méthode	Entrée	Résultats des tests
Déplacement	avancerDe()	distance = 20 cm	Le robot avance précisément de 20 cm.
	reculerDe()	distance = 20 cm	Le robot recule précisément de 20 cm.
	pivoterGauche()/Droite()	angleDeRotation = 90°	Le robot pivote précisément de 90° vers la gauche/droite
	stop()	Aucun	Le robot s'arrête immédiatement.
	allerVersOrientation()	angleCible = 90°	Le robot atteint précisément l'orientation cible.
	orienterVersLigneAdverse()	Aucun	Le robot s'oriente correctement vers la ligne adverse.
Pince	ouvrir()	angle = 1500°	La pince s'ouvre complètement
	fermer()	angle = 1800°	La pince se ferme correctement
	recupererPalet()	Détection d'un palet	La pince attrape le palet avec succès.
	lacherPalet()	Aucun	Le palet est relâché avec succès.
ColorSensor	detecteLigneBlanche()	Lectures de couleur	Retourne true pour une ligne blanche et false sinon.
	close()	Aucun	Le capteur est désactivé correctement.
MoveScan	getDistance()	Entrée capteur	La distance mesurée correspond à la distance réelle.
	forward()	distance = 70 cm	Le robot avance sans collision et s'arrête lorsque nécessaire.
	RechercheEtOrientation()	Aucun	Le robot pivote et identifie correctement les objets.
	moveForward()	Aucun	Le robot avance et s'arrête au point d'arrêt souhaité.
	rotateMeasure()	Aucun	La liste des distances reflète coorectly les objets autour.
	isPalet()	Liste des distances	Les palets sont détectés correctement.
	analyzeDiscontinuity()	Données de discontinuité	L'analyse retourne correctement Palet ou Mur.
MainClass	goToWhiteLineAndDrop()	Aucun	Le palet est déposé avec succès et le robot se réoriente.
	goToSecondPalletAndPick()	Aucun	Le robot récupère le second palet avec précision.

	searchAndCollectNextPallet	Aucun	Le robot localise et collecte les palets de manière autonome.
	detectObstacle()	Aucun	Les obstacles sont détectés de manière fiable.
	eviterObstacle()	Aucun	Le robot contourne les obstacles avec succès.

4. Tests d'Intégration

Test	Description	Procédure de test	Résultat final
Déplacement et détection de ligne blanche	Interaction entre Deplacement et ColorSensor.	1. Placer le robot sur une surface avec une ligne blanche. 2. Lancer goToWhiteLineAndDrop(). 3. Observer si le robot s'arrête sur la ligne blanche.	Le robot s'arrête dès que la ligne blanche est détectée.
Recherche et orientation vers un palet	Interaction entre MoveScan et Deplacement.	1. Placer un palet à 50 cm devant le robot. 2. Lancer RechercheEtOrientation() 3. Observer si le robot s'oriente vers le palet.	Le robot détecte le palet et s'oriente pour s'en approcher.
Collecte et dépôt de palet	Interaction entre Pince, Deplacement, et ColorSensor.	1. Placer un palet à 30 cm devant le robot. 2. Lancer goToWhiteLineAndDrop(). 3. Observer si le palet est collecté, transporté et déposé sur la ligne blanche.	Le palet est déposé sur la ligne blanche.
Évitement d'obstacles	Interaction entre MoveScan et Deplacement.	1. Placer un obstacle à 40 cm devant le robot. 2. Lancer detectObstacle() puis eviterObstacle(). 3. Observer si le robot ajuste sa trajectoire pour éviter l'obstacle.	Le robot contourne l'obstacle et continue son déplacement.
Détection et classification	Interaction entre MoveScan et Deplacement.	1. Placer un obstacle à 40 cm devant le robot. 2. Lancer detectObstacle() puis eviterObstacle(). 3. Observer si le robot ajuste sa trajectoire pour éviter l'obstacle.	Le robot contourne l'obstacle et continue son déplacement.

5. Tests Fonctionnels

Test	Description	Procédure de test	Résultat attendu
Navigation autonome	Le robot navigue en ligne droite, évite les obstacles.	1. Placer le robot sur le terrain. 2. Placer un obstacle à 40 cm. 3. Lancer moveForward(). 4. Observer sa trajectoire et sa capacité à éviter l'obstacle.	Le robot suit une trajectoire droite, détecte et évite l'obstacle pour continuer son déplacement.
Collecte et dépôt de palets	Le robot collecte, transporte et dépose un palet sur une ligne blanche.	1. Placer un palet à 50 cm devant le robot. 2. Lancer goToWhiteLineAndDrop(). 3. Observer les étapes de collecte, transport et dépôt du palet.	Le robot collecte, transporte et dépose le palet à ±5 cm près de la ligne blanche.
Recherche de palet	Le robot détecte un palet, s'oriente, et s'arrête à 10 cm.	1. Placer un palet à 60 cm. 2. Lancer RechercheEtOrientation(). 3. Observer si le robot s'oriente et s'arrête à 10 cm du palet.	Le robot détecte le palet, s'oriente vers lui et s'arrête à une distance de ±2 cm.
Évitement d'obstacles	Le robot détecte un obstacle, ajuste sa trajectoire et continue.	1. Placer un obstacle à 40 cm. 2. Lancer eviterObstacle(). 3. Observer si le robot contourne l'obstacle et reprend sa trajectoire.	Le robot contourne l'obstacle avec une distance de sécurité de ±5 cm et continue son déplacement.
Détection et classification	Le robot détecte et classe des objets (palet, mur, robot).	1. Placer un palet, un mur, et un autre robot à des distances variables. 2. Lancer analyzeDiscontinuity(). 3. Observer les résultats de classification.	Le robot identifie les objets et effectue l'action appropriée avec une précision de classification ≥90%.

6. Conclusion

Ce plan de tests a validé les principales fonctionnalités du robot, telles que la navigation, l'évitement d'obstacles, la collecte et le dépôt de palets, ainsi que la classification d'objets. Les cas testés ont confirmé le bon fonctionnement des composants et identifié les ajustements nécessaires pour améliorer leur précision et leur fiabilité. Ces résultats assurent que le robot répond aux attentes définies.