

Plan de tests



Projet de robotique

Plan de tests

Projet de Robotique

Les informations d'identification du document

Référence du document :	Plan de tests
Version du document :	1.0
Date du document :	16/12/24
Auteur(s) :	Alexander OSTLE Thomas BEGOTTI Victor CHARREYRON

Les éléments de vérification du document

Validé par :	Alexander OSTLE Thomas BEGOTTI Victor CHARREYRON
Validé le :	16/12/24
Soumis le :	16/12/24
Type de diffusion :	Document électronique (.pdf)
Confidentialité :	Standard / Étudiants de l'Université Grenoble-Alpes

Sommaire

Sommaire	2
Introduction	3
Objectifs des méthodes	3
Guide de lecture.....	3
Maîtrise d'œuvre	3
Maîtrise d'ouvrages	3
Concepts de base	3
Tests unitaires.....	3
Test unitaire 1 : Contrôle des moteurs de déplacement.....	3
Test unitaire 2 : Contrôle du moteur des pinces.....	4
Test unitaire 3 : Fonctionnement du capteur tactile.....	4
Test unitaire 4 : Mesure de distance par le capteur ultrasonique	5
Test unitaire 5 : Fonctionnement du capteur couleur	5
Tests d'intégration.....	5
Test d'intégration 1 : Intégration entre les Actionneurs et les Sensors	5
Tests fonctionnels	6
Test fonctionnels 1 : Vérification de la méthode premierPalet.....	6
Test fonctionnels 2 : Vérification de la méthode trouverPalet.....	6
Test fonctionnels 3 : Vérification de la méthode manipulerPalet	7
Vérification de la documentation	7
Glossaire.....	7
Référence	8
Index.....	8

Introduction

Ce plan de tests a pour objectif de vérifier le bon fonctionnement du robot LEGO EV3 développé à l'aide de la bibliothèque LeJOS à travers différents tests.

Objectifs des méthodes

Les tests porteront sur les modules principaux, à savoir les capteurs, les moteurs, et le contrôle du robot en interaction avec ces éléments.

Guide de lecture

Maîtrise d'œuvre

Les développeurs et les testeurs devront suivre ce document pour s'assurer de la conformité des implémentations avec les spécifications et d'effectuer les vérifications nécessaires.

Maîtrise d'ouvrages

L'utilisateur final pourra consulter ce plan pour comprendre les tests effectués sur les différentes fonctionnalités du robot et vérifier leur conformité avec les exigences du cahier des charges.

Concepts de base

Ce document se base sur l'architecture du robot LEGO EV3, comprenant des capteurs (ultrasonique, tactile, couleur) et des moteurs pour le mouvement et la manipulation d'objets.

Tests unitaires

Test unitaire 1 : Contrôle des moteurs de déplacement

- **Identification** : testDéplacement
- **Description** : Vérifier si les moteurs de déplacement (moteur gauche et droit) du robot fonctionnent correctement pour avancer, reculer, et tourner.
- **Contraintes** : L'environnement de test doit être dégagé pour garantir un mouvement sans obstacle.

- **Dépendances** : Les actionneurs doivent être initialisés correctement dans la classe Actionneurs.
- **Procédure de test** :
 1. Faire avancer le robot sur 500 mm.
 2. Faire reculer le robot sur 500 mm.
 3. Faire tourner le robot de 90°.
 4. Vérifier que les actions sont exécutées comme prévu.

Test unitaire 2 : Contrôle du moteur des pinces

- **Identification** : testPinces
- **Description** : Vérifier si le moteur des pinces du robot fonctionne correctement pour ouvrir et fermer les pinces.
- **Contraintes** : L'environnement de test doit être dégagé pour garantir un mouvement sans obstacle.
- **Dépendances** : Les actionneurs doivent être initialisés correctement dans la classe Actionneurs.
- **Procédure de test** :
 1. Ouvrir les pinces avec une vitesse de 700 mm/s.
 2. Fermer les pinces avec une vitesse de 700 mm/s.
 3. Vérifier que les actions sont exécutées comme prévu.

Test unitaire 3 : Fonctionnement du capteur tactile

- **Identification** : testCapteurTactile
- **Description** : Vérifier que le capteur tactile réagit correctement au contact.
- **Contraintes** : Aucun obstacle n'est autorisé autour du capteur tactile pour éviter des fausses lectures.
- **Dépendances** : Les capteurs doivent être initialisés correctement dans la classe Sensors.
- **Procédure de test** :
 1. Appuyer sur le capteur tactile.
 2. Vérifier que le robot détecte le contact (retour "true").
 3. Retirer le doigt et vérifier que le robot détecte l'absence de contact (retour "false").

Test unitaire 4 : Mesure de distance par le capteur ultrasonique

- **Identification** : testCapteurUltrasonique
- **Description** : Vérifier que le capteur ultrasonique fournit des mesures de distance correctes.
- **Contraintes** : L'environnement de test doit permettre de positionner un objet à différentes distances (par exemple, 50 cm, 1 mètre).
- **Dépendances** : Le capteur ultrasonique doit être initialisé dans Sensors.
- **Procédure de test** :
 1. Positionner un objet à 50 cm.
 2. Vérifier que la mesure renvoyée par le capteur est proche de 0.5 mètre.
 3. Répéter l'opération à différentes distances (1 mètre, 2 mètres).

Test unitaire 5 : Fonctionnement du capteur couleur

- **Identification** : testCapteurCouleur
- **Description** : Vérifier que le capteur couleur fournit des couleurs correctes.
- **Contraintes** : L'environnement de test doit permettre de positionner des lignes de couleurs différentes (par exemple rouge, vert, bleu).
- **Dépendances** : Le capteur couleur doit être initialisé dans Sensors.
- **Procédure de test** :
 1. Positionner les lignes de couleurs.
 2. Vérifier que l'entier renvoyé par le capteur correspond à la bonne couleur (à savoir 2 pour bleu, 3 pour vert, 5 pour rouge, les autres couleurs peuvent être retrouvées sur la documentation LeJOS).
 3. Répéter l'opération pour plusieurs couleurs

Tests d'intégration

Test d'intégration 1 : Intégration entre les Actionneurs et les Sensors

- **Identification** : TestInteg1

- **Description** : Tester si le robot réagit correctement lorsqu'il détecte un obstacle à l'aide du capteur ultrasonique et s'arrête avec la commande d'arrêt des moteurs.
- **Contraintes** : L'obstacle doit être placé à une distance de 50 cm du capteur ultrasonique.
- **Dépendances** : Les classes Actionneurs et Sensors doivent être correctement initialisées.
- **Procédure de test** :
 1. Placer un obstacle à 50 cm du capteur ultrasonique.
 2. Lancer le robot en mouvement.
 3. Vérifier que le robot s'arrête correctement lorsque l'obstacle est détecté à la distance spécifiée.

Tests fonctionnels

Test fonctionnels 1 : Vérification de la méthode premierPalet

- **Identification** : testPremierPalet
- **Description** : Tester la méthode premierPalet pour s'assurer que le robot gère bien le premier palet
- **Contraintes** : L'environnement de test doit être identique à la table de jeu.
- **Dépendances** : testDeplacement, testPinces et testCapteurTactile
- **Procédure de test** :
 1. Appeler testpremierPalet.
 2. Vérifier que le robot avance jusqu'au palet le prend dans ses pinces, l'amène à la zone d'en but adverse et le dépose et viens se recentrer sur la table du côté de la zone d'en but adverse.

Test fonctionnels 2 : Vérification de la méthode trouverPalet

- **Identification** : testTrouverPalet
- **Description** : Tester la méthode trouverPalet pour s'assurer que le robot gère bien la recherche d'un palet.
- **Contraintes** : L'environnement de test doit être identique à la table de jeu.
- **Dépendances** : testDeplacement et testCapteurUltrasonique
- **Procédure de test** :
 1. Placer le robot en sachant que qu'il regarde 0°.
 2. Placer un palet à moins de 80 cm derrière le robot.

3. Appeler testTrouverPalet.
4. Vérifier que le robot tourne sur lui-même et affiche en console la distance et l'orientation du palet

Test fonctionnels 3 : Vérification de la méthode manipulerPalet

- **Identification** : testManipulerPalet
- **Description** : Tester la méthode manipulerPalet pour s'assurer que le robot gère bien la manipulation de palets
- **Contraintes** : L'environnement de test doit être identique à la table de jeu.
- **Dépendances** : testDeplacement, testPinces et testCapteurTactile, testCapteurUltrasonique.
- **Procédure de test** :
 1. Centrer le robot sur la table du côté de la zone d'en but et orienter le perpendiculaire à cette zone
 2. Placer un palet dans l'axe centrale de la table derrière le robot à moins de 80cm.
 3. Appeler testManipulerPalet.
 4. Vérifier que le robot se tourne bien vers le palet, va le récupérer, se déplace vers le mur puis longe le mur pour venir le déposer dans la zone d'en but adverse.

Vérification de la documentation

Ce plan de test a été vérifié pour s'assurer qu'il correspond aux exigences du cahier des charges et que tous les tests nécessaires ont été pris en compte.

Glossaire

Méthode : En Java, une méthode est un bloc de code qui réalise une tâche précise ou effectue une action, et qui peut être appelé à plusieurs endroits dans un programme. Les méthodes permettent de structurer et de réutiliser le code, facilitant ainsi la gestion et la maintenance des applications.

Test d'intégration : un test d'intégration désigne le processus de vérification et de validation des différents composants d'un système ou d'un robot, après qu'ils ont été

assemblés ou intégrés. Le but est de s'assurer que ces composants fonctionnent correctement ensemble, en interagissant de manière fluide et efficace. Cela inclut à la fois des tests logiciels et matériels.

Test unitaire : Un test unitaire est une technique de test où une unité individuelle de code, généralement une méthode, est testée de manière isolée. L'objectif principal est de vérifier que chaque petite partie du programme fonctionne correctement indépendamment des autres parties.

Test fonctionnels : Un test fonctionnel est un type de test qui évalue les fonctionnalités d'un système ou d'une application par rapport à ses spécifications et exigences. L'objectif principal est de vérifier si le système fonctionne comme prévu en testant ses fonctionnalités visibles et les actions qu'il est censé effectuer.

True/False : Dans de nombreux langages de programmation, y compris Java, les valeurs true et false appartiennent au type boolean. Ce type sert à représenter deux états opposés (vrai/faux, actif/inactif, 1/0, etc.). Ce type est aussi utilisé en logique.

Référence

- [Cahier des charges du projet](#)
- [La documentation interne du code](#)
- [Documentation LeJOS](#)
- [Plan de développement](#)

Index

Méthode : pages 3,6,7.

Test d'intégration : page 5.

Test unitaire : pages 3,4,5.

Test fonctionnels : pages 6,7.

True/False : page 4.