

Plan de test

Groupe L2A1



Robot ramasseur de palets

Robot ramasseur de palets

Les informations d'identification du document

Référence du document:	A1
Version du document:	1.0
Date du document:	
Auteur(s):	<ul style="list-style-type: none"> • GUENANA Massinissa • KOUADRI Yanis • TANDOU Joh • YE Victor

Les éléments de vérification du document

Validé par :	
Validé le :	
Soumis le :	
Type de diffusion :	Document électronique (.Odt)
Confidentialité :	Standard / Étudiants UFR Maths-Info de l'université Paris Descartes

Année universitaire: 2019/2020

Sommaire:

Table des matières

Sommaire:.....	3
1. Introduction:.....	4
1.1. Objectifs et méthodes.....	4
1.2. Documents de référence.....	4
2. Guide de lecture.....	4
2.1. Maîtrise d'ouvrage.....	4
2.2. Maîtrise d'œuvre.....	4
3. Tests fonctionnels :.....	5
3.1.1. Scenario00:.....	5
3.1.2. Scenario01:.....	5
3.1.3. Scenario02:.....	6
3.1.4. Scenario03:.....	7
3.1.5. Scenario04:.....	9
3.1.6. Scénario05:.....	10
3.1.7. Scenario06:.....	11
3.1.8. Scenario07:.....	12
Tests unitaires:.....	13
4. Tests d'intégration:.....	17
4.1. Test1:.....	17
4.2. Test2:.....	17
4.3. Test3:.....	17
4.4. Test4:.....	17
5. Glossaire.....	18
6. Références.....	18
7. Index.....	19

1.Introduction:

1.1. Objectifs et méthodes

Ce document a pour objectif de décrire l'ensemble de tests et d'activités prévus pour la réalisation de notre projet, qui est de faire fonctionner un robot de modèle LEGO MINDSTORMS EV3, et garantir que le produit réalisé respecte le cahier des charges.

1.2. Documents de référence

1. Cahier des charges
2. Cahier de recette
3. Conception détaillée

2.Guide de lecture

2.1. Maîtrise d'ouvrage

Le maître d'ouvrage définit les besoins du cahier des charges. Dans le cadre de notre projet, c'est le client. Le descriptif des besoins et fonctionnalités ainsi que la planification du projet est utile au client.

2.2. Maîtrise d'œuvre

Le maître d'œuvre est la personne ou entreprise chargée de la conception du projet, il s'agit ici de développeurs chargés de programmer le robot (les étudiants).

3. Tests fonctionnels :

3.1.1. Scenario00:

Identification: P0

Description: Le robot doit pouvoir saisir un palet lorsqu'il le détecte.

Contrainte: Aucune

Dépendance: Aucune

Procédure de test:

Situation initiale	Situation finale
Le robot est déposé au hasard sur la table.	Le robot détecte le palet et ferme ses pinces.

Validation du test: Le test est validé si le robot a fermé ses pinces au contact du palet.

3.1.2. Scenario01:

Identification: P1

Description: Le robot doit pouvoir donner la couleur de la ligne sur laquelle il a été déposé. A savoir que la table est composée de 7 couleurs (deux lignes blanches délimitant le terrain, la surface grise, et les lignes rouge, jaune, noire, verte et bleue).

Contrainte: Le robot doit prendre en compte les changements d'éclairages pour que les résultats ne soit pas faussés.

Dépendance: Aucune.

Procédure de test:

Situation initiale	Situation finale
Le robot est déposé au hasard sur une des lignes de la table.	Le robot indique la couleur de la ligne sur laquelle il a été déposé.

Validation du test: Le test est validé si la couleur de la ligne est celle affichée par le robot.

3.1.3. Scenario02:

Identification: P2

Description: Ce scenario a pour but d'observer les mouvements du robot. Le robot doit pouvoir trouver un palet placé sur une des 9 intersections de la table, le ramasser et le déposer derrière la ligne blanche du camp adverse.

Contrainte: Le robot doit utiliser le capteur de couleurs pour savoir dans quel camp il se trouve.

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
Un palet est déposé au hasard sur une des 9 intersections de la table.	Le robot franchit la ligne blanche du camp adverse avec le palet, s'arrête et ouvre ses pinces.
Le robot est déposé au hasard sur l'une des trois positions de départ du camp (Est ou Ouest) qui lui a été désigné.	

Validation du test: Ce test est validé si le robot a saisi puis déposé le palet derrière la ligne blanche du camp adverse.

3.1.4. Scenario03:

Identification: P3

Description: Ce scenario a pour but d'observer les mouvements du robot. Le robot doit pouvoir trouver un palet placé sur une des 9 intersections de la table, le ramasser et le déposer derrière une des deux lignes blanches.

Contrainte: Aucune

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
--------------------	------------------

Un palet est déposé au hasard sur une des 9 intersections.	Le robot franchit une des deux lignes blanches avec le palet, s'arrête et ouvre ses pinces.
Le robot est déposé au hasard n'importe où sur la table excepté sur une ligne.	

Validation du test: Ce test est validé si le robot a saisi puis déposé le palet derrière une des deux lignes blanches.

3.1.5. Scenario04:

Identification: P4

Description: Ce scenario a pour but d'observer les mouvements du robot. Le robot doit pouvoir trouver un palet placé sur une des 9 intersections de la table, le ramasser et le déposer derrière la ligne blanche du camp adverse.

Contrainte: Le robot doit utiliser le capteur de couleurs pour savoir dans quel camp il se trouve.

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
Le camp adverse est désigné au robot: Est ou Ouest.	Le robot franchit la ligne blanche du camp adverse avec le palet, s'arrête et ouvre ses pinces.
Un palet est déposé au hasard sur une des 9 intersections de la table.	
Le robot est déposé au hasard n'importe où sur la table exceptée sur une ligne.	

Validation du test: Ce test est validé si le robot a saisi puis déposé le palet derrière la ligne blanche du camp adverse.

3.1.6. Scénario05:

Identification: P5

Description: Ce scenario a pour but d’observer les mouvements du robot. Le robot doit pouvoir trouver les 9 palets placés sur les 9 intersections de la table, les ramasser et les déposer derrière la ligne blanche du camp adverse.

Contrainte: Le robot doit utiliser le capteur de couleurs pour savoir dans quel camp il se trouve.

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
Le camp adverse est désigné au robot: Est ou Ouest.	Le robot franchit la ligne blanche du camp adverse avec chacun des palets, s’arrête et ouvre ses pinces.
Les 9 palets sont déposés sur les 9 intersections de la table.	
Le robot est déposé au hasard sur l’une des trois positions de départ du camp qui lui a été désigné.	

Validation du test: Ce test est validé si le robot a saisi puis déposé chacun des 9 palets derrière la ligne blanche du camp adverse.

3.1.7. Scenario06:

Identification: P6

Description: Ce scenario a pour but d'observer les mouvements du robot. Le robot doit pouvoir trouver un palet placé dans une zone grise de la table, le ramasser et le déposer derrière une des deux lignes blanches.

Contrainte: Aucune

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
Un palet est déposé au hasard n'importe où sur la table, excepté sur une des 9 intersections de la table ou sur une ligne.	Le robot franchit une des deux lignes blanches avec le palet, s'arrête et ouvre ses pinces.
Le robot est déposé au hasard n'importe où sur la table, excepté sur une ligne.	

Validation du test: Ce test est validé si le robot a saisi puis déposé le palet derrière une des deux lignes blanches.

3.1.8. Scenario07:

Identification: P7

Description: Ce scenario a pour but d'observer les mouvements du robot. Le robot doit pouvoir trouver un palet placé dans une zone grise de la table, le ramasser et le déposer derrière la ligne blanche du camp adverse.

Contrainte: Le robot doit utiliser le capteur de couleurs pour savoir dans quel camp il se trouve.

Dépendances: Aucune

Procédure de test:

Situation initiale	Situation finale
Le camp adverse est désigné au robot: Est ou Ouest.	Le robot franchit la ligne blanche du camp adverse avec le palet, s'arrête et ouvre ses pinces.
Un palet est déposé au hasard n'importe où sur la table, excepté sur une des 9 intersections de la table ou sur une ligne.	
Le robot est déposé au hasard n'importe où sur la table, excepté sur une ligne.	

Validation du test: Ce test est validé si le robot a saisi puis déposé le palet derrière la ligne blanche du camp adverse.

Tests unitaires:

Description:

Vérifier et tester les classes et méthodes citées ci-dessous:

- **Brique:**

Cette classe concerne la brique ev3 et contient une méthode **choisirCamp()** qui permet de saisir le camp du robot.

- **Capteur Photosensible:**

Cette classe concerne le capteur de couleurs et contient la fonction permettant au robot de détecter les couleurs.

1. **initialisation():** initialise le capteur de couleurs.
2. **obtenirIDCouleur():** scanne le point se situant sous le capteur photosensible et renvoie l'id (int compris entre -1 et 13) de la couleur détectée.
3. **fermer1():** libère les ressources du capteur de couleurs.

- **Capteur Tactile:**

Cette classe concerne le capteur de contact et contient une méthode **estPresse()** qui permet de détecter une pression.

1. **fermer2():** libère les ressources du capteur de contact.

- **Capteur Ultrason:**

Cette classe concerne le capteur ultrason et contient les fonctions permettant la détection d'un objet ainsi que la mesure de sa distance par rapport au robot :

1. **activation():** active le capteur à Ultrason.
2. **obtenirDistance():** évalue la distance séparant le robot de l'obstacle le plus proche.
3. **detecterObjet():** détermine si un objet est proche, et s'il s'agit d'un palet ou du robot adverse.
4. **desactivation():** désactive le capteur ultrason.
5. **fermer4():** libère les ressources du capteur ultrason.

- **UltrasonicClosest:**

Cette classe sert à détecter si un autre objet s'y trouve à une distance minimale, et à un degré précis. Elle a plusieurs méthodes:

1. **ultrasonicproche(float degre, float taille):** elle sert à détecter un objet à un degré précis.
2. **detecter(float Dmin):** elle sert à déterminer si le robot détecte un objet à moins d'une distance minimale.

- **Moteur:**

Cette classe concerne les trois moteurs et contient les fonctions permettant la mise en mouvement du robot. Elle a trois variables d'instance de type UnregulatedMotor (moteur gauche, moteur droit, pince). Elle a plusieurs méthodes :

1. **départ():** le robot avance droit jusqu'à trouver la ligne rouge ou jaune.
2. **avancer1():** le robot avance droit jusqu'à ce qu'il détecte le vide ou une ligne.
3. **avancer2(int a):** le robot suit une ligne à allure rapide jusqu'à ce qu'il détecte une couleur donnée.

4. **avancer3(int n):** le robot suit une ligne en effectuant un nombre donné d'oscillations.
5. **avancer4(int a):** le robot suit une ligne à une allure lente jusqu'à ce que le robot détecte une couleur donnée.
6. **avancer5():** le robot se déplace jusqu'à détecter la ligne jaune.
7. **avancer5Bis():** le robot se déplace jusqu'à détecter la ligne rouge.
8. **reculer1():** le robot recule en effectuant une oscillation sur une ligne.
9. **reculer2():** le robot recule droit jusqu'à ce qu'il détecte une ligne.
10. **tournerGauche(int a):** le robot tourne vers la gauche jusqu'à ce qu'il détecte couleur donnée.
11. **tournerDroite(int a):** le robot tourne vers la droite jusqu'à ce qu'il détecte couleur donnée.
12. **faireDemiTour(int a):** le robot tourne vers la droite jusqu'à ce qu'il détecte une couleur donnée.
13. **allerIntersection(int i, boolean x, boolean camp):** le robot se déplace jusqu'à une intersection donnée.
14. **ramasserPalet(int i):** le robot saisit un palet et le déplace jusqu'à la ligne blanche adverse pour le déposer.
15. **fermer3():** libère les ressources des moteurs.

Contraintes :

- Le robot doit être sur la surface jeu.
- Le robot doit être suffisamment chargé.

Procédures de test:

Données en entrée: les classes et méthodes.

Résultats attendus:

1. Reconnaître les couleurs
2. Détecter un palet
3. Saisir un palet
4. Déposer un palet dans le camp adverse
5. Se déplacer vers le palet le plus proche
6. Se repérer dans l'espace (reconnaître le camp adverse)
7. Reconnaître le robot adverse
8. Eviter tout contact avec le robot adverse

4. Tests d'intégration:

4.1. Test1:

Description: ce test sert à vérifier la cohérence entre les moteurs (moteurs Gauche, moteurs Droit, les pinces) et le capteur tactile.

Résultat: au contact d'un palet, le capteur tactile le détecte, ce qui provoque l'arrêt des moteurs. Le moteur pince fait ouvrir la pince et permet le ramassage du palet.

4.2. Test2:

Description: ce test sert à vérifier la cohérence entre les moteurs et le capteur à Ultrason.

Résultat: au fur et à mesure que le robot se déplace, le capteur à Ultrason calcule la distance le séparant d'un autre objet (palet, mur). Si ce dernier détecte un objet à une distance inférieure à 20 cm, le programme fait actionner les moteurs du robot (faire demi-tour, avancer ...).

4.3. Test3:

Description: ce test sert à vérifier la cohérence entre le capteur de couleurs et les moteurs.

Résultat: au fur et à mesure que le robot se déplace, le capteur de couleurs détecte la couleur de la ligne sur laquelle il se déplace.

4.4. Test4:

Description: vérifier la cohérence entre l'ensemble des classes du programme.

Résultat: toutes les classes interagissent entre elles et chacune influence le fonctionnement de l'autre.

5. Glossaire

- **Capteur:** dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable.
- **Java:** langage de programmation orienté objet.
- **Lejos EV3:** minuscule machine virtuelle java.

6. Références

http://www.ens.math-info.univ-paris5.fr/projets-informatiques/doku.php?id=projets:documentation#dokuwiki__top

<https://fr.wikipedia.org/wiki/Capteur>

<http://www.lejos.org/ev3.php>

7.Index

Index lexical

capteur de couleurs.....	7
la brique ev3.....	7