

PLAN DE TESTS



ASSIKIDANA Esdras
BOULLI Marouan
GUYOT Alizée
LAURENT-PAOLI Pierre
MEHDI Khadidja

2020

Sommaire :

Introduction.....	1
I. Tests unitaires	1
1. Package Action	1
A. Classe Déplacement.....	1
B. Classe Pince.....	3
2. Package Perception	4
A. Classe UltrasonicSensor	4
B. Classe TouchSensor.....	5
C. Classe ColorSensor	6
II. Tests d'intégration	7
1. Rechercher un palet proche et le saisir	7
2. Déposer un palet derrière la ligne blanche	7
3. Se calibrer à l'aide du mur adverse	8
4. Suivre le circuit	8

Introduction

Ce document présente les différents tests effectués pour vérifier le bon fonctionnement du robot, et des stratégies.

I. Tests unitaires

1. Package Action

A. Classe Déplacement

Test avancer(double distance)

1) Description :

Faire avancer le robot de 20 centimètres en avant. Mesurer la distance parcourue et la comparer avec la distance attendue.

Répéter 5 fois.

Résultats :

Le robot avance d'une distance comprise entre 19 et 21 centimètres. La précision n'est pas totalement exacte.

2) Description :

Faire avancer le robot d'une distance aléatoire comprise entre 0 et 100 centimètres. Mesurer la distance parcourue et la comparer avec la distance attendue.

Répéter 5 fois.

Résultats :

Plus la distance à parcourir augmente, plus la distance réellement parcourue est inexacte. Pour les distances inférieures à 50 centimètres, la marge d'erreur est de 1 à 2 centimètres. Pour des distances supérieures à 50 centimètres, la marge d'erreur peut aller jusqu'à 3 ou 4 centimètres.

Test reculer(double distance)

1) Description :

Faire reculer le robot de 20 centimètres en arrière. Mesurer la distance parcourue et la comparer avec la distance attendue.

Répéter 5 fois.

Résultats :

Pour deux essais, le robot recule d'une distance comprise entre 19 et 21 centimètres. La précision est parfois exacte mais ne l'est pas toujours.

2) Description :

Faire reculer le robot d'une distance aléatoire comprise entre 0 et 100 centimètres. Mesurer la distance parcourue et la comparer avec la distance attendue.

Résultats :

Plus la distance à parcourir augmente, plus la distance réellement parcourue est inexacte. Pour les distances inférieures à 50 centimètres, la marge d'erreur est de 1 centimètres. Pour des distances supérieures à 50 centimètres, la marge d'erreur peut aller jusqu'à 2 centimètres.

Conclusion :

Les tests pour les méthodes de Déplacement : avancer et reculer, ont démontré qu'il y a une différence entre les distances fournies dans le code et les distances réellement parcourues. Cette différence peut être négligeable si l'on considère que le robot a d'autres moyens d'évaluer les distances, notamment grâce aux lignes de couleurs, aux palets et aux murs.

Test turnRight(double angle)

1) Description:

Faire tourner le robot d'un angle compris entre 0 et 360°. Afficher l'angle parcouru sur l'écran du robot et comparer avec l'angle réellement parcouru.

Répéter 5 fois.

Résultats :

Les mesures faites à l'aide d'un rapporteur sont exactement les mêmes que les mesures attendues.

2) Description :

Effectuer plusieurs rotations sur le robot jusqu'à le faire revenir à sa position initiale.

Répéter 5 fois.

Résultats :

À chaque essai, le robot ne revient pas complètement à sa position initiale mais avec un petit décalage.

Test turnLeft(double angle)

Résultats :

Les tests effectués pour cette méthode sont les mêmes que ceux de la méthode **turnRight(double angle)**. Les résultats obtenus sont aussi identiques.

Tests finaux

1) Description :

Pour tester toutes les méthodes, effectuer un parcours qui fait tourner le robot de différents angles, le fait avancer et reculer de différentes distances. Le robot doit être de retour à sa position initiale à la fin de ce parcours.

Répéter 2 fois.

Résultats :

Le robot ne revient jamais réellement à sa position initiale, mais à une position plus ou moins identique selon le nombre de rotations et de parcours effectués.

2) Description :

Effectuer le parcours précédent mais avec des obstacles.

Répéter 2 fois.

Résultats :

La position n'est plus du tout exacte quand le robot entre en contact avec des obstacles.

Conclusion :

La différence entre la position attendue et la position réellement atteinte n'est plus du tout négligeable quand le robot entre en contact avec des obstacles. Il est donc nécessaire de programmer le robot pour qu'il détermine une nouvelle position initiale grâce à un mur. Cela peut se faire de plusieurs manières : soit au bout d'un temps défini (par exemple au bout d'une minute), soit au bout d'un nombre de rotations défini, soit à chaque but marqué.

B. Classe Pince

Test 1

Description :

Tester la fermeture et l'ouverture des pinces du robot sans palet. Modifier le nombre de tours pour des valeurs allant jusqu'à 1000, et vérifier si elles se ferment entièrement.

Vérifier si le booléen **aPalet** vaut bien false.

Résultats : Le booléen vaut bien false car le TouchSensor n'a pas été stimulé. Pour le premier point, 1000 tours suffisent à ouvrir les pinces du robot suffisamment grand. Sans palet, les pinces se ferment entièrement pour 1000 tours (valeur qui est définie pour l'ouverture des pinces au début de la partie).

Test 2

Description :

Tester la fermeture et l'ouverture des pinces du robot avec un palet. Modifier le nombre de tours pour des valeurs allant de 500 à 1000, par tranche de 100.

Vérifier si le booléen **aPalet** vaut bien true.

Résultats :

Le booléen vaut true car le TouchSensor a été stimulé. Il faut bien que le palet se trouve au milieu et appuie sur le TouchSensor. Pour la fermeture des pinces, 900 tours suffisent à fermer suffisamment pour attraper le palet. Pour des valeurs supérieures à 900, les pinces forceront trop sur le palet et peuvent se casser. Pour des valeurs inférieures à 900, les pinces ne seront pas assez fermées pour tenir correctement le palet et il est possible de le perdre.

Conclusion :

Il faut définir 1000 tours pour la première ouverture des pinces. Par la suite, 900 tours suffisent pour ouvrir et fermer les pinces tout au long du jeu. Il reste cependant le cas où à cause d'un mur ou d'une collision, les pinces se déboitent et il ne sera pas possible de les remettre en place. Il sera donc utile d'appeler un temps mort si l'on en a le droit.

2. Package Perception

A. Classe UltrasonicSensor

Test getDistance()

Description :

Ce test permet de savoir si le capteur d'ultrasons évalue correctement les distances.

- 1) Placer un palet en face du robot à plus de 32 cm. Afficher la distance retournée et vérifier qu'il s'agit de la bonne distance en mesurant la distance réelle entre le robot et le palet.
- 2) Placer un palet en face du robot à moins de 32 cm et un autre derrière à plus de 32 cm. Afficher la distance retournée et vérifier qu'il s'agit bien de la distance séparant le robot du deuxième palet.
- 3) Placer le robot en face d'un mur à moins de 32 cm. Afficher la distance retournée et vérifier qu'il s'agit bien de la bonne distance.

Répéter 5 fois. (Avec 5 distances différentes pour chaque étape)

Résultats attendus :

- 1) Le capteur doit retourner une distance proche de la distance réelle le séparant du palet.
- 2) Le capteur retourne la distance séparant le robot du second palet.
- 3) Le capteur doit retourner une distance proche de la distance réelle le séparant du mur.

Test detectPalet()

Ce test a pour but de vérifier que le robot détecte les palets.

Description :

Placer un palet à plus de 32 cm en face du robot. Faire avancer le robot et lui dire de s'arrêter lorsqu'il détecte le palet. S'il s'arrête c'est qu'il a bien détecté le palet, sinon c'est qu'il ne l'a pas détecté et il faut alors stopper le programme.

Répéter 5 fois en plaçant le palet à des distances différentes à chaque fois.

Résultats attendus :

Le robot s'arrête lorsqu'il détecte le palet (lorsqu'il est à environ 30 cm du palet).

Test detectWall()

Ce test permet de vérifier que le robot détecte les murs.

Description :

Placer le robot en face d'un mur à plus de 32 cm, le faire avancer et lui dire de s'arrêter lorsqu'il détecte le mur. S'il s'arrête c'est qu'il a bien détecté le mur, sinon stopper le programme car il n'a pas détecté le mur. Placer un palet sur la trajectoire pour vérifier qu'il ne confond pas les deux.

Répéter 5 fois.

Résultats attendus :

Le robot s'arrête lorsqu'il détecte le mur (lorsque le capteur retourne une distance inférieure à 30 cm)

B. Classe TouchSensor

Test de validité du TouchSensor

Description :

Ce test permet de savoir si le TouchSensor est en état de fonctionner correctement. Pour cela, il suffit d'ouvrir la fenêtre EV3 Control et enfoncer manuellement le TouchSensor pour voir si la valeur rendue (true ou false) correspond bien à l'état du TouchSensor.

Répéter 5 fois.

Résultats attendus :

La fenêtre EV3 renvoie true quand le bouton est enfoncé et false quand il ne l'est pas.

Test IsPressed()

Description :

Appuyer manuellement sur le TouchSensor pour voir si son état correspond à ce qu'il renvoie sur la console du robot.

Répéter 5 fois.

Résultats attendus :

La console renvoie true quand le bouton poussoir est enfoncé et false quand ce dernier ne l'est pas.

Test aEteTouche()

Description :

Placer un robot en face d'un palet et le faire avancer jusqu'à ce que le palet rentre en contact du TouchSensor.

Répéter 5 fois.

Résultats attendus :

Le robot s'arrête dès qu'il détecte que le bouton poussoir a été enfoncé. Si le robot ne s'arrête pas, alors son TouchSensor n'a jamais été simulé, ce qui signifie qu'il ne détecte pas le palet.

C. Classe ColorSensor

Test getCurrentColor()

Ce test a pour but de vérifier que le robot détecte correctement les couleurs.

Description :

Placer le capteur du robot sur chaque ligne de couleur.
Changer de position et la luminosité (si possible).

Résultats attendus :

La couleur retournée à l'écran doit être conforme à la couleur de la ligne sur lequel le robot se trouve.

Test changeColor()

Ce test sert à vérifier que le robot détecte les changements de couleur lorsqu'il se déplace sur le terrain.

Description :

Placer le robot sur une couleur (de préférence sur les parties grises du terrain) et le laisser avancer jusqu'à la rencontre d'une autre couleur (blanche, bleu, etc.).

Surveiller la couleur affichée sur l'écran du robot lors de ses déplacements.

Répéter 5 fois

Résultats attendus :

Le robot affiche la bonne couleur lorsqu'il y a un changement de couleur.

II. Tests d'intégration

1. Rechercher un palet proche et le saisir

But du test :

Vérifier que le robot est capable d'effectuer une recherche, d'identifier le palet le plus proche et de le saisir.

Principe de réalisation :

- 1) Placer plusieurs palets tout autour du robot, tous à plus de 32 cm du robot mais à des distances différentes.
- 2) Effectuer une rotation de 360 degrés tout en prenant des mesures de distances.
- 3) Effectuer une nouvelle rotation d'un angle correspondant à la distance la plus proche mesurée lors de la première rotation.
- 4) Ouvrir les pinces et avancer jusqu'à toucher le palet.
- 5) Une fois le palet touché, fermer les pinces pour le saisir.

Résultats attendus :

Le robot doit saisir le palet le plus proche.

2. Déposer un palet derrière la ligne blanche

But du test :

Vérifier que le robot est capable de s'arrêter et de lâcher le palet une fois la ligne blanche détectée.

Principe de réalisation :

- 1) Placer le robot perpendiculairement à la ligne blanche et mettre un palet entre ses pinces fermées.
- 2) Avancer jusqu'à détecter la ligne blanche et ouvrir les pinces pour déposer le palet

Résultats attendus :

Le robot doit déposer le palet derrière la ligne blanche.

3. Se calibrer à l'aide du mur adverse**But du test :**

Vérifier que le robot se calibre correctement à l'aide du mur (à la fin de l'opération il doit être perpendiculaire au mur)

Principe de réalisation :

- 1) Placer le robot sur la ligne blanche en face du mur adverse.
- 2) Effectuer un balayage (d'environ 50 degrés = 25 degrés à gauche et 25 degrés à droite) tout en prenant des mesures de distance.
- 3) Effectuer une rotation d'un angle correspondant à la plus petite distance mesurée

Résultat attendu :

À la fin de l'opération, le robot doit être perpendiculaire au mur.

4. Suivre le circuit**But du test :**

Vérifier que le robot suive correctement le circuit, qu'il récupère les palets lorsqu'il les trouve et les dépose dans le but adverse, qu'il passe au point suivant du circuit s'il ne trouve pas de palet et qu'il se repositionne s'il ne touche pas le palet malgré l'avoir détecté. Pour rappel, le robot est censé avancer d'une distance légèrement supérieure à celle le séparant du palet lorsqu'il veut le saisir.

Principe du test :

- 1) Placer le robot sur la ligne de départ et les palets à leurs positions respectives.
- 2) Faire suivre le circuit au robot et récupérer les palets.
- 3) Retirer un palet du circuit pour vérifier que le robot passe au point suivant lorsqu'il ne trouve pas de palet.
- 4) Pour un palet, le décaler légèrement après que le robot l'ait détecté et qu'il avance pour le saisir, pour vérifier qu'il se repositionne lorsqu'il ne touche pas le palet malgré l'avoir détecté.

Résultats attendus :

Le robot doit suivre le circuit, récupérer les palets lorsqu'il les trouve et les déposer dans le but adverse.

Lorsqu'il ne trouve pas de palet, il doit se diriger vers le point suivant.

Si après avoir détecté un palet il se dirige vers lui mais ne le touche pas, il doit reculer, effectuer une nouvelle recherche et retenter sa chance. S'il ne le touche toujours pas, il doit passer au point suivant.