

## **Sommaire:**

<b>Test Unitaire</b>	<b>2</b>
CU (Capteur Ultrason)	2
Mise en oeuvre	2
Résultats	2
CCC (Calibration Capteur Couleur)	2
Mise en oeuvre	2
Résultats	2
CC (Capteur Couleur)	3
Mise en oeuvre	3
Résultats	3
MR (Moteurs Roues)	3
Mise en oeuvre	3
Résultats	3
CT (Capteur Tactile)	3
Mise en oeuvre	3
Résultats	3
MP (Moteur Pince)	4
Mise en oeuvre	4
Résultats	4
<b>Test d'intégration</b>	<b>4</b>
Robot	4
Description	4
Principe de réalisation	4
RoutineCompétition	4
Description	4
But du test	5
Principe de réalisation	5

Ce document est une liste exhaustive et une explication de tous les tests qui ont été menés lors du projet en intelligence artificielle. Il y a les tests unitaires et également les tests d'intégration. Les tests recensent à la fois les tests pour les classes moteurs et capteurs, qui sont des classes outils mais également les tests des classes intégrées, c'est à dire les classes qui se servent des classes outils pour fonctionner.

## Test Unitaire

### I. CU (Capteur Ultrason)

#### a. Mise en oeuvre

Pour tester la classe CU, nous avons réalisé la classe TestCU. Pour exécuter cette classe, il faut se placer dans un environnement simple avec une règle, une salle épurée. Cette classe va permettre de vérifier que le capteur ultra-son du robot situé sur les yeux estime comme il faut la distance. Pour tester la fonction `getDistance`, il faut exécuter la classe en mettant d'abord le robot devant un objet proche (15/20 cm), puis devant un objet éloigné (1m), puis enfin devant aucun objet devant lui (il ne détecte pas de distance supérieur à 2m). Pour le test de la fonction `getLastDistance` vous n'avez pas besoin de placer le robot dans un environnement spécifique; il vous vérifiera que le `getLastDistance` retourne la même valeur que le précédent test de `getDistance`.

#### b. Résultats

Pour vérifier que les tests fonctionnent, comparer la valeur que le robot indique entre la distance des yeux et de l'objet que vous avez placés en mesurant avec votre règle la distance. On remarque que la distance est approximée, il y a quelques centimètres de différence. On remarque également que lorsque le robot est placé trop loin des objets, il renvoie la valeur `infinity`. Enfin, on remarque que dans un angle, le robot affiche une distance non conforme. La méthode `getLastDistance` fonctionne car les données concordent avec le précédent résultat de `getDistance`.

### II. CCC (Calibration Capteur Couleur)

#### a. Mise en oeuvre

Afin de tester la calibration du capteur couleur, il suffit de réaliser le calibrage et de vérifier que le fichier est bien créé.

#### b. Résultats

Après la calibration, le fichier est bien créé.

### III. CC (Capteur Couleur)

#### a. Mise en oeuvre

Afin de vérifier que la calibration est correcte, nous avons fait une classe vérifiant chaque couleur. Ainsi, il faut placer le robot sur la ligne de couleur à tester, appuyer sur le bouton et vérifier que la couleur captée est bien celle de la ligne.

#### b. Résultats

Nous obtenons une calibration correcte, sauf parfois pour la ligne blanche que le robot détecte comme du jaune, lorsque ce dernier a un palet dans les pinces et est en mouvement. Afin de palier à ce problème, lorsque le robot détecte du jaune, celui-ci s'arrête et effectue un second test.

### IV. MR (Moteurs Roues)

#### a. Mise en oeuvre

Pour tester la classe MR nous avons réalisé la classe TestMR. Il suffit de l'exécuter sur le robot lejos et d'actionner n'importe quel bouton pour lancer le programme de test.

#### b. Résultats

Il faut appuyer sur n'importe quel bouton pour passer à l'étape suivante, une instruction va s'afficher à l'écran du robot, le test est réussi si le robot effectue ce qu'il y a marqué sur son écran, sinon il faut corriger les paramètres de la classe.

### V. CT (Capteur Tactile)

#### a. Mise en oeuvre

Pour tester la classe capteur tactile nous avons réalisé la classe CT-test. Il suffit d'appuyer sur le bouton et de vérifier sur le robot que le message "appuie détecter" apparaisse.

#### b. Résultats

Si le message s'affiche c'est que le bouton a correctement perçu l'appuie. On remarque que les vérifications d'appuie sont sur des délais très courts il se peut donc qu'un appuie très court ne soit pas perçue.

## VI. MP (Moteur Pince)

### a. Mise en oeuvre

Pour tester cette classe nous avons créé la classe MP-test. Cette classe teste chacune des ouvertures/fermetures de la pince. Grâce aux tachocount on peut observer sur le robot le nombre de tour qui ont été effectués.

De plus pour trouver le coefficient entre l'ouverture et la fermeture nous avons testé environ 40 fois en prenant les valeurs obtenues grâce au compteur de tour du moteur.

### b. Résultats

Si le nombre de tour effectué à des valeurs équivalentes à l'aller et au retour et que la pince n'a pas perdu sa symétrie c'est que cela fonctionne. Il est possible que le manque de précision du robot fasse que l'ouverture et la fermeture n'est pas le même nombre de tour pour être égal.

## Test d'intégration

### I. Robot

#### a. Description

La classe Robot dispose de 5 classes de test (TestRotationRecherche, TestDeposerPaletEnButEtAjustementDPEB, TestPremierPalet, TestRétablirPince et TestRetablirPince), ces dernières

ont été conçues pour tester une ou deux méthodes à la fois (explicitée dans le nom des classes de test), indépendamment des autres méthodes. Les méthodes sont toutes testées ensemble dans la classe RoutineCompetition\_test.

#### b. Principe de réalisation

Chaque classe possède une boucle qui permet de tester la ou les méthodes plusieurs fois. Il suffit d'appuyer sur le bouton central du robot pour (re)lancer la méthode et d'appuyer sur le bouton du bas pour mettre fin au programme. Pendant l'exécution de chaque classe il faut vérifier si le robot effectue, ce qui a été programmé dans la ou les méthodes concernées.

### II. RoutineCompetition

#### a. Description

La classe RoutineCompetition\_test permet d'observer à tout moment l'avancée du robot et les choix réalisés.

b. But du test

Ce test permet de comparer facilement les prédictions du constructeur et ce que réalise le robot en contrôlant chacun de ses choix et de ses effets.

c. Principe de réalisation

À chaque fois que l'automate change d'état, ce dernier est affiché et il faut appuyer sur le bouton central pour poursuivre le programme ou appuyer sur le bouton du haut pour l'interrompre.