Projet IA & Robotique

Plan de tests

Tuteur:

DAMIEN Pellier

Étudiants:

AURAY Cédric

CALVIGNAC Sébastien

SIMON Dorian

KAMISSOKO Djoko

Année scolaire : 2020-2021

Sommaire

I) Introduction

Le plan de test regroupe l'ensemble des tests de fonctionnalités et d'intégration. Il décrit en détail la fonctionnalité en question, ses contraintes physiques, ses dépendances à d'autres méthodes et la procédure de test. La procédure de test décrit en langage courant la méthodologie du test à réaliser.

II) Test fonctionnels

Pilote

prendre

Description
Fais fermer la pince de 600 (unités arbitraires).
Contraintes
Robot. Moteur de pince.
Dépendances
Lejos API.
Procédure de test
Lancer la commande et vérifier que la pince ferme mais pas plus que la taille du palet pour éviter le grincement.

lacher

Description
Fais fermer la pince de 600 (unités arbitraires).
Contraintes
Robot. Moteur de pince.
Dépendances
Lejos API.
Procédure de test
Lancer la commande et vérifier que la pince ouvre correctement.

reglerVitesse

Description
Défini la vitesse de déplacement en fonction des roues gauche et droite.
Contraintes
Robot. Moteur de roues.
Dépendances
Lejos API et notamment la méthode setSpeed de l'interface Chassis. Avancer.
Procédure de test
Faire avancer le robot avec des valeurs de vitesse distinctes pour vérifier qu'il y a changement. Et vérifier que l'avancement à vitesse maximale ne pose pas de problème de précision de déplacement pour les autres méthodes.

rotate

Description
Permet au robots de tourner d'un angle donné en paramètre tout en mettant à jour l'attribut rotation.
Contraintes
Robot.
Dépendances
Lejos API, notamment lejos.robotics.navigation.MovePilot.rotate.
Procédure de test
Faire plusieurs rotations et vérifier que l'angle entré en paramètre reflète la réalité. Finir par tourner à un angle égale à l'inverse de la valeur stockée dans l'attribut rotation (historique des angles) pour vérifier que le robot se tourne bien de nouveau vers l'angle de départ.

Capteurs

actualise

Description
Cette méthode met à jour la distance perçue par le capteur d'ultrasons et stocke le résultat dans un attribut de la classe.
Contraintes
Avoir accès au rebot et à la salle de compétition
Dépendances
Classes ultrason, Touche, Couleur. Calibrage. API lejos.
Procédure de test
Le test consiste à faire afficher la distance, le toucher, et la couleur perçue par le robot sur l'écran du robot toutes les secondes tout en déplaçant le robot. Il faut s'assurer que les valeurs affichées sont cohérentes, c'est-à-dire que la distance, couleur et couleur de l'objet en face du robot affiché reflète bien la réalité.

Couleur

scalaire

Description
Cette méthode prend en paramètres deux tableaux de 3 float chacun contenant les valeurs correspondant virtuellement aux valeurs RGB échantillonnées via le capteur de couleurs.
Contraintes
échantillon de 7 couleur distinctes, calibrage réalisé.
Dépendances
méthode actualise() de la classe couleur
Procédure de test
Instancier une classe capteur qui va instancier une classe couleur. La calibration est réalisée via l'appel du constructeur de Couleur. Réaliser un affichage en temps réel via une boucle de l'attribut this.couleur qui sera mis à jour par appel répété à actualise();

actualise

Description
Permet de récupérer un nouvel échantillon RGB sous forme d'un tableau de 3 float, pour le comparer ensuite aux données enregistrées dans les tableaux attributs portant le nom des couleurs virtuellement calibrées (tableaux de 3 float également) de l'instance Couleur. Par l'évaluation de conditionnelles successives l'on va mettre à jour l'attribut this.couleur selon la différence minimale évaluée entre notre échantillon courant et les différents échantillons attributs. La différence minimale est évaluée et enregistrée si inférieure à la précédente dans une variable double minscal. Si la nouvelle valeur obtenue sur appel de scalaire() est inférieure à une valeur précédemment enregistrée, this.couleur est actualisée pour représenter sous forme d'une String le nom de l'attribut avec lequel l'échantillon courant est comparé. Aucun retour.
Contraintes
Calibration réalisée sur un ensemble de 7 échantillons suffisamment distincts en termes de valeurs RGB.
Dépendances
scalaire(), fetchSample(), attributs de l'instance.
Procédure de test
Dans une boucle sur un main, appel continu à actualise() et affichage en temps réel de this.couleur. L'on porte le capteur couleur sur différentes surfaces correspondantes en termes de couleurs aux matériaux utilisés durant la calibration.

Ultrason

actualise

Description
Cette méthode met à jour la distance perçue par le capteur d'ultrasons et stocke le résultat dans un attribut de la classe.
Contraintes
Avoir le robot en présence
Dépendances
API lejos
Procédure de test
Le test consiste à faire afficher la distance perçue par le robot sur l'écran du robot toutes les secondes tout en déplaçant un objet devant le robot. Il faut s'assurer que les valeurs affichées sont cohérentes, c'est-à-dire que la distance de l'objet en face du robot affiché reflète bien la réalité.

Touche

actualise

Description
Cette méthode met à jour le contact perçu par le capteur de toucher et stocke le résultat en dans un attribut de la classe.

Contraintes
Avoir le robot en présence

Dépendances
API lejos

Procédure de test
Le test consiste à faire afficher le contact perçu par le robot sur l'écran du robot toutes les secondes tout en actionnant le bouton de toucher. Il faut s'assurer que les valeurs affichées sont cohérentes, c'est-à-dire que l'écran affiche "0" quand le bouton toucher n'est pas appuyé et inversement "1" quand il l'est.

III) Tests d'intégration

deplacementAleatoire

Description
Cette méthode permet de déplacer le robot aléatoirement selon une valeur de temps.Elle permet de se déplacer selon une valeur de temps aléatoirement généré entre 0 et 10000 Ms,puis d'effectuer une rotation d'une valeur générée aléatoirement entre 0 et 360° si le capteur de distance perçoit une valeur inférieure à ~20cm ou que le capteur couleur détecte une ligne blanche, on effectue une rotation de 180° avant de sortir de la méthode si le capteur de pression est activé durant le déplacement, on met à jour l'attribut palet Saisi et l'on sort de la méthode également, on met à jour l'attribut derniereLigneTraversee si l'on perçoit une ligne de couleur vert ou bleu via le capteur couleur.
Contraintes
Robot. Table de compétition.
Dépendances
Lejos API. Les méthodes reglervitesse, rotate et forward de Pilote. la méthode getcouleur de capteurs.
Procédure de test
Placer le robot sur la table de compétition, lancer la méthode. S'assurer que le robot est capable de sortir des angles, éviter les mûres et ne dépasse jamais la ligne blanche.

chercherPalet

Description
Cette méthode permet de chercher les palets sur le plateau.
Contraintes
Table de compétition.
Dépendances
deplacementAleatoire, Touche, Capteurs,Pilote.
Procédure de test
Placer des palets sur la table de compétition et vérifier que le robot s'arrête devant un palet et non pas un mûre par exemple.

prendrePalet

Description
Cette méthode permet de prendre le palet avec la pince en supposant que le palet se trouve devant lui.
Contraintes
Robot. Palet.
Dépendances
Lejos API.
Procédure de test
Placer un palet proche devant le robot et lancer la méthode. La méthode devra retourner true si le palet a été pris et false si'il rentre en contacte avec une ligne blanche ou bien qu'il avance d'une distance maximale de 20cm.

marquerPalet

Description
Cette méthode est appelé lorsque le robot a saisit un palet. Cette méthode se sert de l'attribut rotation (historique des angles) pour s'orienter vers la base adverse.
Contraintes
Table de compétition.
Dépendances
Lejos API. rotation
Procédure de test
placer le robot sur la table en direction de la zone d'en-but avec les pinces en position fermé et avec un palet dedans. Faire plusieurs rotations. lancer la méthode. On s'attend a ce que le robot repose le palet dans la zone adverse.

marquerPalet2

Description
Comme pour la méthode précédente, celle-ci souhaite placer le palet saisi dans la zone d'en but. Mais la difference c'est que cette méthode ne dépend pas de l'attribut rotation (historique des angles). Au lieu elle se sert des couleurs des lignes et notamment il se souvient de couleur de la ligne adverse et donc vérifie que celle-ci est présente proche de la ligne blanche.
Contraintes
table de compétition
Dépendances
Lejos API. chercherBut.
Procédure de test
Placer le robot sur la table avec les pinces en position fermé et avec un palet dedans. Initialiser les attributs couleurDepart et couleurArrivee pour refléter la réalité. CouleurDepart correspond à la couleur de la première ligne parallèle en face de la zone de départ et inversement pour couleurArrivee. puis lancer la méthode. On s'attend à ce que le robot fassse des déplacements aléatoires jusqu'à rencontrer la ligne blanche de l'adversaire dans quel cas il vérifie que la ligne d'arrivée est correcte (en se déplaçant vers elle) puis dépose le palet dans la zone d'en-but

chercherBut

Voir marquerPalet2

calibrer

Description
Méthode qui permet de mettre à jour les valeurs de l'attributs rotation. Cette méthode permet au robot de se remettre "droit" (c'est à dire perpendiculaire au mur)
Contraintes
On a besoin d'une surface plane / mur.
Dépendances
On a besoin de la classe pilote pour pouvoir faire bouger le robot et de la classe Capteur pour pouvoir interroger le sonar.
Procédure de test
Pour tester cette procédure, on met le robot de travers en face d'un mur. Si le robot arrive à se mettre perpendiculaire au mur c'est que la méthode marche de façon correcte.

Homologation

Description
méthode permettant d'avancer jusqu'à récupérer un palet, le saisir, effectuer une légère rotation, avancer jusqu'à l'en-but adverse et déposer le palet. Si la méthode échoue la méthode est abandonné après une rotation de 180°
Contraintes
Robot. Ruban blanc. Palet.
Dépendances
Cette méthode dépend du bon fonctionnement de la classe pilote et de la classe Capteur qui dépend du fonctionnement de Touche, de Couleur et d'Ultrason
Procédure de test
Placer un palet au moins 30cm devant le robot, puis une ligne blanche perpendiculaire à la direction de départ et à au moins 1m du robot, qui fait office de zone d'en-but.

Séquences Principales
Pour les séquence 1, 2, 3 et 4 voir respectivement les test d'intégrations pour homologation, déplacementAleatoire, prendrePalet et marquerPalet.

Le test d'intégration finale consiste à mettre le robot dans les conditions de compétition, c'est-à-dire sur la table de compétition, avec tous les palets et avec un adversaire.