



L3 MIASHS - S5
Septembre 2024

Programmation d'un robot LEGO Mindstorms

Cahier des charges

BOUCQ-KIEFFEL Léna	12202198
DIALLO Abdoul Khadir	12217447
DUBUS Roxane	12207968

Table des matières

1. Introduction.....	3
a) Contexte.....	3
2. Description de la demande.....	3
a) Objectifs.....	3
b) Produit du projet.....	3
c) Fonctions du produit.....	3
3. Contraintes.....	4
a) Contraintes de délais.....	4
b) Contraintes matérielles.....	4
c) Autres contraintes.....	4
4. Déroulement du projet.....	5
a) Planification.....	5
b) Ressources.....	5
c) Organisation.....	5

1. Introduction

a) Contexte

Une course de robots est organisée à la fin du semestre dans le cadre du cours "Initiation à l'intelligence artificielle". Le but de cette course est que le robot récupère le plus de palets possibles disposés sur un terrain de jeu et les dépose dans une zone d'en-but dans un temps imparti de 2 min 30. Il sera face à un autre robot, qui aura les mêmes consignes. Le robot doit être programmé de manière efficace afin qu'il soit rapide dans ses mouvements et adaptatif face aux situations imprévisibles, comme par exemple un robot adverse qui lui bloque le chemin. Cette course sera évaluée et les gagnants auront des points supplémentaires sur la note finale.

2. Description de la demande

a) Objectifs

Le code devra programmer le robot de manière à ce qu'il puisse démarrer, scanner son environnement, attraper des palets et les ranger. Le robot devra, dans un temps limité, obtenir le plus rapidement possible un maximum de palets et il devra adapter sa trajectoire en fonction de l'autre robot présent sur le plateau. Le plus dur sera de faire fonctionner efficacement le robot, même dans des situations inattendues.

b) Produit du projet

La stratégie mise en place commence par se placer sur la table toute à gauche ou toute à droite en fonction du robot adverse, afin de ne pas être en vis-à-vis. De préférence, le robot sera du côté gauche, afin que s'il rencontre quand même le robot adverse, il tourne à droite vers l'intérieur du plateau, et non à gauche vers le mur. Il sondera la couleur où il se trouve grâce au capteur de couleurs. Le robot ira prendre le premier palet devant lui, afin de limiter l'accès de palets proches au robot adverse, quand celui-ci se retrouvera en face. Une fois le palet attrapé, le robot tournera soit à droite, soit à gauche selon l'emplacement où il a commencé, afin de se mettre à mi-chemin entre deux lignes et de ne pas bousculer les autres palets. Le robot sera programmé de manière à ce qu'il tourne toujours vers l'intérieur du plateau (dans la limite du possible). Le robot aura un compteur de degrés (initialisé à 0), afin de savoir dans quelle direction il roule par rapport à la longueur du plateau. Le compteur augmentera de 90 degrés si le robot tourne à droite, et diminuera de 90 s'il tourne à gauche. Si le compteur de degrés est de 0 et qu'il sonde la ligne blanche, la pince s'ouvrira, car il sera dans la zone d'en-but et il pourra lâcher le palet.

c) Fonctions du produit

Le code se constitue de 4 classes. La première classe "Déplacement" aura pour but de déplacer le robot. Les méthodes seront "tournerDroite()", "tournerGauche()", "avancer(int distance)" et reculer(int distance). Les méthodes permettent au robot de tourner à droite ou à gauche, ainsi que d'avancer et de reculer d'une distance d (en mètre). De plus, il y aura une fonction calculPaletPlusProche(), le robot sondera son environnement avec le capteur de couleur et avancera vers le palet le plus proche de lui. La classe "Détection" permettra de programmer les différents capteurs du robot. Il y aura une méthode pour le capteur "couleur()", qui détectera les couleurs afin de savoir où le robot se trouve et quelles actions faire. Par exemple, s'il capte la ligne blanche et qu'il sait qu'il a tourné de 0 degré, il est dans la zone d'en-but et peut lâcher le palet. Un autre capteur sera "tactile()", quand

celui-ci est activé, le robot refermera ses pinces, car il saura qu'il a trouvé un palet. Les méthodes "ultrason()" et "son()" aideront le robot à détecter l'autre robot, ainsi que les palets et les différents obstacles sur son chemin. La classe "Mouvement" permettra de prendre et poser un palet à l'aide des méthodes "prendre()" et "poser()" et du capteur tactile. La dernière classe sera une classe qui donnera la distance, la couleur de la ligne sur laquelle le robot se trouve (avec le capteur de couleurs) et un angle pour tourner. Le robot captera aussi s'il se dirige vers le mur avec le capteur d'ultrason. Une fois arrivé près de la zone d'en-but adverse, il essaiera toujours d'aller chercher le palet le plus proche et de le ramener.

3. Contraintes

a) Contraintes de délais

Des livrables intermédiaires sont à rendre tout au long de la réalisation du projet avec les échéances suivantes :

- Semaine 3 (23/09) : cahier des charges
- Semaine 5 (07/10) : plan de développement
- Semaine 10 (18/11) : plan de test
- Semaine 11 (25/11) : code source et documentation interne
- Semaine 12 (02/12) : rapport final

b) Contraintes matérielles

Pour ce projet, nous devons utiliser un robot LEGO Mindstorms, avec deux moteurs pour le faire avancer et une pince également motorisée pour attraper les palets. Ce robot est équipé des capteurs suivants :

- un capteur tactile,
- un capteur de couleur,
- un capteur ultrason,
- un capteur de son.

Le robot se déplacera sur un plateau avec des lignes de couleurs pour l'aider à se diriger. De plus, les batteries devront être suffisamment chargées pour couvrir la durée complète de chaque match.

c) Autres contraintes

D'autres contraintes devront être respectées dans la réalisation de ce projet. Le programme du robot devra être fait en Java avec l'utilisation de la bibliothèque LeJOS pour faire fonctionner les capteurs et les moteurs. Un dépôt Git devra être utilisé pour la gestion du code et pour assurer un suivi collaboratif. De plus, lors des matchs, le robot doit obligatoirement éviter toute collision avec le robot de l'équipe adverse.

4. Déroulement du projet

a) Planification

Pour ce projet, nous avons établi six étapes principales qui seront à suivre durant sa conception. Tout d'abord, les étapes primordiales sont l'écriture du cahier des charges puis, la mise en place du plan de développement avant de commencer la rédaction du programme. Ensuite, pour structurer notre manière de programmer sans perdre de temps, il faudra établir le plan de tests qui auront lieu au fur et à mesure de l'écriture du programme.

Une fois ces tâches terminées, nous pourrons commencer la conception du programme. Cette phase est divisée en deux étapes : en premier, écrire les classes et méthodes dites "basiques" comme avancer, tourner, etc. Ces méthodes seront utiles à la deuxième étape qui est d'écrire les classes et méthodes qui utilisent les méthodes "basiques" et mettant en place la stratégie.

Enfin, en fonction des tests que nous ferons au fur et à mesure, il faudra adapter et modifier le programme ou/et la stratégie selon les résultats des tests.

De plus, tout le long du projet aura lieu la phase de rédaction du rapport pour qu'il soit complété en fonction du déroulement du projet.

b) Ressources

Afin de mener à bien ce projet, plusieurs ressources sont à notre disposition. La documentation que l'on pourra enrichir à l'aide de nos recherches est primordiale pour établir une stratégie et plus particulièrement pour écrire le programme. Ensuite, pour pouvoir rédiger le programme et l'exécuter, le logiciel Eclipse sera utilisé. De plus, un GitHub du projet est créé pour faciliter le partage de la documentation et de l'écriture du code. Enfin, les ressources matérielles principales sont les palets, le plateau sur lequel le robot concourra et le robot accompagné de ses capteurs.

c) Organisation

Afin de s'organiser dans le temps, un échéancier a été créé avec les différentes tâches à faire tout au long du semestre. L'échéancier aidera à avoir une vue d'ensemble de l'avancée du projet et à gérer le temps. Il sera mis à jour en fonction des tâches ajoutées/modifiées.

Dans une première phase de planification, le groupe rédige le cahier des charges, l'échéancier et le plan de développement. Le groupe fera les tests ensemble au Fablab à MUSE.

Dans la seconde phase, chaque membre du groupe programme une ou plusieurs classes "basiques" du robot, afin d'avoir une bonne base pour continuer.

Ensuite, la phase de développement et de test commencera, ou le groupe étoffera son programme et sa stratégie, et fera des tests au fur et à mesure afin de s'assurer du bon fonctionnement et de trouver la meilleure stratégie possible, ainsi que de limiter les imprévus. A ce moment commencera aussi la rédaction du rapport.

Pour l'organisation de la communication dans le groupe du projet, la plateforme Whatsapp est utilisée pour partager les informations générales et pour discuter de l'avancée générale et de chacun.es sur le projet. Lorsqu'il le sera nécessaire, nous nous rencontrerons lors de réunions afin de mieux discuter du projet. Les séances de TP serviront également de lieu de discussions pour l'organisation.