Année 2022-2023

Projet Intelligence Artificielle

Cahier des charges

Université Grenoble Alpes - L3 MIASHS

Théo Laigle Délyan Zergua

Table des matières

[Introduction 3](#_Toc114131915)

[I. Description de la demande 3](#_Toc114131916)

[II. Contraintes 5](#_Toc114131917)

[III. Déroulement 6](#_Toc114131918)

# Introduction

Ce projet s’inscrit dans le cadre du module Intelligence Artificiel de la L3 MIASHS. Celui-ci nous propose de découvrir les bases de l’intelligence artificielle en programmant un petit robot Lego sous Java. Nous sommes lancés dans une compétition avec les autres étudiants dans laquelle nos robots se mesureront les uns aux autres au problème du ramassage de palets. Nous sommes répartis en 8 groupes de 2 à 5 étudiants. L’ensemble des TP et cours auront lieu au laboratoire Fablab sur le Campus de Saint-Martin-d’Hères et seront dispensés par M. Damien Pellier.

# I. Description de la demande

L’objectif du projet est de programmer un robot en vue de participer à une compétition. Dans cette compétition, il y a 8 groupes et donc 8 robots. Ces derniers s’affrontent en 1 contre 1 sur une table de jeu de 3 mètres par 2 mètres. Cette aire de jeu est entourée d’une bordure rigide d’une hauteur de 15 centimètres pour rendre impossible la sortie des robots. Le cœur de la table est quadrillé par 6 lignes de couleurs (2 noires, 1 verte, 1 bleue, 1 jaune et 1 rouges). Parmi ces lignes, 3 sont disposées horizontalement et les 3 autres sont disposées verticalement. Au sein de ce quadrillage, il y a 9 intersections, qui représentent les 9 emplacements où des palets seront placés.

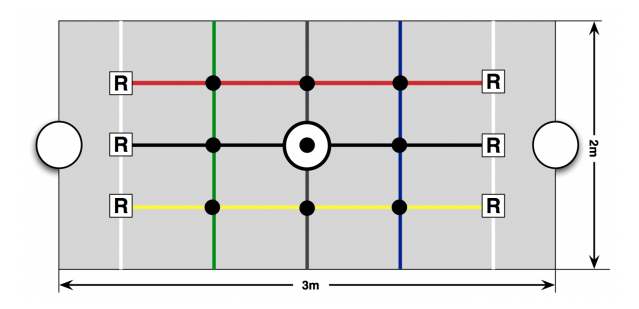


Figure : Plan de la table de jeu.

Au début de chaque affrontement, les 2 robots se trouvent à chacune des extrémités de la table. Chaque robot peut choisir d’où il souhaite démarrer tant qu’il ne dépasse pas la ligne blanche que l’on voit sur la figure 1. Durant le match, le but pour les robots est d’apporter les palets dans la zone adverse (au-delà de la ligne blanche). Pour gagner un match, il y a un système de points. Le calcul des scores s’effectue comme suit :

* 5 points pour chaque palet placé dans la zone d’en-but adverse
* 3 points de bonus pour l’équipe ayant marqué en premier
* 2 points pour le robot ayant un palet en sa possession à la fin de la manche

Il n’y a pas seulement le fait de placer des palets dans le camp adverse qui rapporte des points. Il peut être intéressant de viser les bonus de 3 points et 2 points pour tenter de gagner.

L’objectif principal du projet est donc de réaliser un programme permettant au robot de récupérer des palets et puis de les amener dans le but adverse. Mais au-delà de cet objectif, mettre toutes les chances de son côté pour battre les autres robots engagés dans la compétition. Pour cela la partie mise en place de stratégies est essentielle pour programmer le robot le plus robuste de la compétition. Par stratégie, on entend mettre en place un maximum d’actions à réaliser par le robot pour lui permettre d’être prêt à affronter chaque situation qu’il peut rencontrer.

Cela se traduit par un algorithme déterministe dans lequel, nous allons coder les différentes actions à réaliser par le robot. A noter que le meilleur moyen d’obtenir un algorithme robuste et de minimiser les erreurs est de proposer, à chaque fois, plusieurs façons de réaliser la même action pour le robot, en utilisant des capteurs différents (caméra, détecteur de couleurs, détecteur à ultrason).

La structure de l’algorithme prendra la forme d’automates, qui seront eux-mêmes basés sur des instructions conditionnelles.

Puis finalement, c’est l’équipe qui aura établi la meilleure stratégie qui gagnera, au du moins qui mettra toutes les chances de son côté. Il faut ainsi imaginer tous les scénarios qui peuvent se produire.

Pour cela, nous disposons d’un robot LEGO Mindstorms EV3, commercialisé en 2013. Avec sa brique moteur “EV3”, ce robot fait partie des derniers modèles commercialisés par LEGO au sein de la gamme Mindstorms.



Figure : brique moteur EV3.

Cette brique “EV3” possède 4 ports pour les moteurs (A,B,C et D) et 4 ports pour les capteurs (1,2,3 et 4). Le téléversement de programme se fait soit par le biais d’une connexion à distance (Bluetooth ou Wi-Fi) soit par un port USB situé sur le côté de la brique.

La brique est équipée d’un microprocesseur 32 bit ARM7 d’ATMEL. Ainsi que d’un écran à cristaux liquides de 100 par 64 pixels.

Les dimensions du robot sont de 11,2 x 7,2 x 4 centimètres.

Grâce à toutes les caractéristiques listées ci-dessus, le robot possède de nombreuses fonctionnalités. Il a des capteurs qui lui permettent de percevoir l'environnement dans lequel ils se trouvent. Il y a :

* Un capteur tactile qui réagit à la pression
* Des capteurs de sons qui permettent de mesurer l’intensité sonore en décibels.
* Des capteurs à ultrasons pour détecter les objets et mesurer des distances (en centimètres ou en pouces). De plus, ils permettent d’identifier d’autres robots grâce à leurs capteurs ultrasons.
* Un capteur de couleurs pour distinguer les différentes couleurs mais aussi calculer l’intensité de la lumière ambiante.

Le robot possède aussi plusieurs moteurs qui lui permettent de faire des actions dans son environnement. Il y a :

* Deux moteurs pour faire tourner ces deux roues et ainsi lui permettre d’avancer ou de reculer.
* Un moteur pour ouvrir ou fermer la pince du robot.

De plus, le robot dispose d’un haut-parleur qui émet des sons et peut être très utile pour donner des indications, plus particulièrement pendant les phases de tests.

A noter qu'au-dessus de l’aire de jeu, il y a une caméra qui risque d’être très utile car elle permet de repérer la position instantanée des robots ainsi que des palets.

Pour mettre toutes les chances de son côté, il faut prendre en compte tous les scénarios non souhaités. Puis de développer un code qui minimise les chances de tomber sur ces scénarios qui peuvent coûter cher.

# II. Contraintes

Parmi les contraintes, on trouve d’abord les différentes échéances :

* Semaine 3 (19/09 – 23/09) : Cahier des charges
* Semaine 5 (03/10 – 07/10) : Plan de développement
* Semaine 10 (14/11 – 18/11) : Plan de Tests
* Semaine 11 (21/11 – 25/11) : Code source et documentation interne
* Semaine 12 (28/11 – 02/12) : Rapport final

Nous ne sommes que deux étudiants dans le groupe, nous avons donc une contrainte supplémentaire par rapport aux autres groupes. Notons aussi que nous n’avons jamais fait de Java avant cette année de L3. Nous avons donc également la contrainte de devoir nous former tout au long du projet.

Les modalités de la compétition nous imposent une contrainte de qualité. Le tournoi final aura un impact sur notre note. Il nous faut donc être bien placés par rapport aux autres groupes. Cette compétition définit d’ailleurs des règles à respecter pour assurer l’équitée entre les groupes. Il est par exemple interdit de piloter manuellement le robot lors d’un match. Celui-ci doit être en autonomie totale. Il est également interdit de modifier physiquement le robot. Les différences entre ces derniers ne peuvent qu’être d’ordre logiciel. Certaines stratégies sont interdites pour éviter la dégradation des robots. Par exemple, il est interdit d’utiliser notre robot comme un bélier pour rendre notre adversaire inopérant.

# III. Déroulement

La première étape consiste à prendre le robot en main, faire fonctionner ses différents capteurs et actionneurs. Une fois cela fait, nous utiliserons ces fonctionnalités dans des cas simples : avancer jusqu’à ce que le palet appuie sur le bouton toucher, tourner sur soit jusqu’à détecter un palet, déplacer le palet vers l’avant, … Puis nous intègreront des nouvelles fonctionnalités au fur et à mesure jusqu’à ce que le robot se retrouve dans une situation comparable à celle des matchs de compétition, mais sans adversaire. Nous élaborerons ensuite des stratégies afin de maximiser nos chances face à un adversaire. Chaque étape sera accompagnée d’une phase de test sur la table de jeu avec des palets. En fin de projet, il conviendra de nettoyer et optimiser le code, puis d’élaborer une documentation.

L’organisation du travail lors du projet sera simple : nous sommes deux dans le groupe, nous travaillerons donc à deux sur les tâches successives. Avec le temps, et si nous voyons que nous devenons plus à l’aise, nous travailleront sur deux tâches chacun de notre côté, puis nous vérifierons le travail réalisé à deux.

Pour atteindre notre objectif, nous coderons trois classes : Une pour les capteurs, une pour les actionneurs, et une classe « agent » qui contiendra l’algorithme du robot.

Voici une première idée d’algorithme :

Notre robot est placé sur l’emplacement de gauche.

* Avancer jusqu’à contact avec le palet
* Fermeture des pinces
* Tourner de 45° sur la gauche puis avancer de quelques centimètres
* Tourner de 45° sur la droite puis avancer jusqu’à détection de la ligne blanche
* Ouverture des pinces puis reculer de quelques centimètres
* Rotation dans le sens horaire jusqu’à détecter d’un palet
* Avancer jusqu’à contact avec le palet
* Rotation jusqu’à faire face au camp adverse (comment savoir quand on fait face au camp adverse ?)
* Avancer jusqu’à ligne blanche
* Répéter les quatre dernières étapes