



# **Cahier des Charges**

## Projet Intelligence Artificielle

Eva DROSZEWSKI  
Sara IKAN  
Anesie MARTINIANI  
Alix PRATABUY

<b>1) Introduction.....</b>	<b>1</b>
a) Contexte.....	1
b) Historique.....	2
<b>2) Description de la demande.....</b>	<b>2</b>
a) Les objectifs.....	2
b) Produit du projet.....	2
c) Les fonctions du produit.....	2
<b>3) Contraintes.....</b>	<b>5</b>
a) Contraintes de délais.....	5
b) Contraintes matérielles.....	5
c) Autres contraintes.....	6
<b>4) Déroulement du projet.....</b>	<b>6</b>
a) Planification.....	6
b) Ressources.....	6
c) Organisation.....	7
<b>5) Glossaire.....</b>	<b>8</b>
<b>6) Annexes.....</b>	<b>9</b>
Annexe I : Diagramme de Gantt.....	9
Annexe II : Dimensions du robot.....	10
<b>7) Références.....</b>	<b>11</b>

# 1) Introduction

## a) Contexte

Dans le cadre d'un projet centré sur l'Intelligence Artificielle, un robot sera programmé dans le but de ramasser un maximum de palets, sur un plateau, en un temps minimal. Après 12 semaines de préparation, le robot sera mis en compétition avec ceux des sept autres équipes, tous étudiants en Licence 3 MIASHS. La préparation ainsi que la compétition se sont déroulées au FABLAB du bâtiment MUSE de l'Université Grenoble Alpes. Le but de ce document est de définir les exigences et les spécifications techniques du projet de programmation du robot. Ce document sert de guide pour structurer la conception, le développement et la mise en œuvre du projet, en détaillant les attentes, les contraintes, et les livrables. Il vise à assurer une compréhension entre les membres de l'équipe, mais aussi avec le commanditaire.

## b) Historique

Le terme d'intelligence artificielle a été inventé par John McCarthy en 1956. McCarthy la définit comme "une Construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus cognitif de haut niveau tels que l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique." Cependant, l'idée remonte à un peu plus loin, avec le Test d'Alan Turing, en 1950, où il se demandait si les machines pouvaient penser avec son article "Computing machinery and intelligence". L'histoire de l'intelligence artificielle connaît un développement rapide depuis les années 50 jusqu'à aujourd'hui. Ce projet a pour objectif d'introduire et de familiariser les étudiants à l'intelligence artificielle.

# 2) Description de la demande

## a) Les objectifs

L'objectif de ce projet est de programmer un robot pour qu'il puisse récupérer un maximum de palets dans un temps imparti. Le langage de programmation utilisé est Java. Le robot doit pouvoir se déplacer, se diriger, détecter et collecter les palets en un minimum de temps. Il doit être capable de se déplacer à gauche, à droite, ainsi qu'en avant et en arrière. Il doit également être capable de détecter les autres robots afin d'éviter les collisions et de minimiser les risques de dommages. La vitesse de la tâche doit être optimisée dans le but d'un gain de temps lors de la compétition. Le robot doit exécuter ses tâches de manière efficace, en optimisant ses déplacements et ses mouvements. Il doit pouvoir manipuler les palets avec précision, sans les perdre, pour les déposer dans la zone précise.

## b) Produit du projet

Les livrables du projet comprennent le code source et la documentation interne, le cahier des charges, le plan de développement, le plan de test ainsi que le rapport final du projet. Ils sont à fournir tout au long de la préparation du projet. Des réunions hebdomadaires sont également prévues afin de faire un point régulier avec le commanditaire Damien Pellier.

### c) Les fonctions du produit

1. Déplacement (avancer et reculer) et orientation (tourner à gauche/ à droite de X de degrés)
  - Avancer et reculer : Le robot avance ou recule grâce aux méthodes `avancer_deS`, `avancer_deA`, `reculer_deS` et `reculer_deA`.
  - Rotation : le robot peut tourner de manière précise grâce aux méthodes `tourner_deS` et `tourner_deA`. La rotation est contrôlée en degrés.
  - Mouvement aléatoire : La méthode `mouvement_aleatoire` permet au robot de tourner de manière aléatoire.
2. Détection et prise de décision
  - Calcul des distances et détection des discontinuités : le robot utilise un capteur à ultrasons (`getDistance`) pour mesurer les distances et les stocker pour ensuite pouvoir identifier les palets et les murs grâce à la détection de la discontinuité des distances en considérant qu'il y a une discontinuité lorsque la différence entre deux mesures de distances successives est supérieure à un seuil prédéfini (`detecterDiscontinuite`).
  - Détection des obstacles : Les obstacles sont détectés et contournés grâce à la méthode `eviter_obstacle`.
3. Manipulation des palets
  - Prendre et déposer : les méthodes `attraper_palet` et `deposer_palet` gèrent automatiquement la prise et le dépôt des palets grâce au moteur dédié pour les pinces.
  - État des pinces : les pinces du robot peuvent s'ouvrir et se fermer via les méthodes `ouvrir_pinces` et `fermer_pinces`, avec une vérification de leur état (`getEstOuvert`).
4. Gestion des états
  - Debut : initialisation des mouvements et des pinces.
  - RechercherPalet : Recherche des palets grâce à un mouvement aléatoire.
  - PrendrePalet : Récupérer le palet identifié.
  - RamnerPalet : Dépose le palet récupéré dans la camp adverse.
  - Fin : Fin des actions lorsque toutes les tâches sont accomplies.

## 3) Contraintes

### a) Contraintes de délais

Nous devons rendre les documents suivants au fil des semaines pour le suivi du projet. Ils sont à déposer sur le wiki du github du projet. Le tableau ci-dessous regroupe les différentes deadlines de chaque document.

Cahier des charges	Semaine 3 : 23/09 - 29/09
Plan de développement	Semaine 5 : 07/10 - 13/10
Plan de tests	Semaine 11 : 25/11 - 01/12
Documentation interne du code + code source	Semaine 12 : 02/12 - 08/12
Date de livraison du produit (concours)	Semaine 12 : 02/12 - 08/12

Pour plus de précision, voir diagramme de Gantt en annexe.

### b) Contraintes matérielles

Pour réaliser ce projet, nous devons utiliser un Robot LEGO Mindstorms Tribots. Les plans de ce robot ne peuvent pas être modifiés, nous devons l'utiliser seulement avec les moteurs et capteurs spécifiés. Il possède des capteurs tactiles, sonores, à

ultrasons et de couleurs ainsi que des capteurs de rotations (servomoteurs). Ils permettent au robot de prendre des mesures qui lui seront nécessaires afin d'interagir avec son environnement. Il est également composé d'une pince, actionnée par un moteur, qui lui permettent d'attraper des palets. Il est doté d'une unité de contrôle programmable qui possède une fonction Bluetooth, lui permettant de se connecter via un ordinateur ou un téléphone portable. Le robot comprend un microprocesseur 32 bit ARM7 d'ATMEL, un port USB, quatre ports d'entrées pour la connexion des capteurs (nommés 1,2, 3, 4) et trois ports de sortie, nommés A, B et C, pour les moteurs. Il est doté d'un écran à cristaux liquides de 100 par 64 pixels et d'un haut-parleur intégré. Les dimensions du robot sont de 112 mm de longueur, 72 mm de largeur et 40 mm de hauteur (voir annexe II). Il est alimenté par six piles AA et une batterie neuf Volt. Ces batteries pourront être remplacées à la fin de chaque manche ou au cours des temps morts. Il est donc crucial de bien gérer leur autonomie afin d'éviter toute panne pendant un match.

Etant donné que le capteur de couleur ne fonctionne pas, nous avons opté pour une autre stratégie afin que le robot puisse se repérer dans la table. Nous avons opté pour une boussole qui permet de définir le nord, sud, est et l'ouest en fonction des différentes valeurs d'angle. On fait une fonction qui permet de reset l'angle à 0 et elle sera utilisée dans la fonction NORD de la boussole.

Les matchs ont lieu sur un terrain prédéfini, de 3m par 2m avec plusieurs zones délimitées par des lignes de couleurs. La couleur de fond de ce terrain est gris clair. Les lignes blanches marquent la limite des en-buts. Les lignes vertes et bleues délimitent respectivement l'Est et l'Ouest du terrain tandis que les lignes rouges et jaunes le Nord et le Sud. Les lignes noires partagent le terrain en son milieu de l'Est à l'Ouest et du Nord au Sud. Les palets à ramasser seront positionnés aux intersections des lignes. La profondeur de l'en-but est de 30 cm et chaque zone du terrain délimitée par les lignes de couleur a une dimension de 50cm x 60cm (voir figure 1).

### c) Autres contraintes

Nous devons nous soumettre aux règles du tournoi. Tout d'abord, le robot subit un test d'homologation sur l'aire de jeu. En effet, pour pouvoir participer à la compétition, le robot doit être capable (en moins de trois minutes) de : soit se déplacer de son point de départ à la zone d'en-but adverse ou de saisir un palet placé au centre du terrain et de la déposer dans la zone d'en-but. Si le robot échoue lors de cette phase, il ne pourra pas participer à la compétition. La deuxième phase consiste à la rencontre de chaque équipe pendant un match. Un match est composé de deux manches d'une durée de cinq minutes chacune avec une pause de cinq minutes entre elles. On peut demander un temps mort si le robot tombe en panne dans les 30 premières secondes d'une manche, mais il n'est pas autorisé dans les phases finales. Au début de chaque manche, les robots sont positionnés sur le repère R, à l'est ou à l'ouest. Les robots doivent collecter les palets à l'aide de leurs pinces et les déposer dans la zone adverse. Seuls les palets déposés et non poussés sont comptabilisés.

De plus, il faut éviter toute collision entre les robots afin de limiter les risques de détérioration ou de casse. Le robot doit donc être conçu de sorte à éviter les autres robots.

Le robot doit être totalement autonome pendant les matchs, on ne peut pas le contrôler à distance, ni lui transmettre des informations en temps réel. Il faut donc programmer en amont toutes les décisions et actions que le robot devra exécuter pendant la compétition.

Contrainte d'organisation : le projet doit être conçu en équipe de quatre. Les rendus devront se faire via la page wiki du github.

## 4) Déroulement du projet

### a) Planification

Les différentes phases du projet ainsi que les étapes qui correspondent sont décrites dans un diagramme de Gantt (voir annexe I).

### b) Ressources

Les ressources humaines du projet sont constituées des membres du groupe (Sara IKAN, Alix PRATABUY, Eva DROSZEWSKI, Anesie MARTINIANI) et du professeur Damien PELLIER, qui encadre le projet. Chaque membre contribue au développement du robot et à la compétition.

Concernant les ressources matérielles, nous disposons d'un robot LEGO Mindstorms Tribot équipé de divers capteurs (tactile, sonore, à ultrasons et de couleurs) ainsi que de servomoteurs qui servent de capteurs de rotation.

Le robot est alimenté par 6 piles AA et est doté d'une fonction Bluetooth permettant la connexion à un PC, d'un microprocesseur 32 bit ARM7 D'ATMEL, de 4 ports d'entrée, 3 ports de sortie, 1 port USB 2.0, un écran ainsi qu'un haut-parleur intégré.

Le développement du code se fera via Eclipse IDE en JAVA, qui permet d'écrire et de tester les programmes sur la brique NXT.

### c) Organisation

Notre groupe est composé de quatre membres comme indiqué précédemment. Afin de garantir une répartition équitable de la charge de travail, nous divisons le groupe en deux équipes, chacune se chargeant du développement et du test du code.

La répartition des tâches dépendra des classes à développer. Pour certaines classes plus simples, nous travaillerons en binômes, permettant ainsi d'avancer plus rapidement sur le développement du code. En revanche, pour les classes les plus complexes, nous travaillerons ensemble tout en répartissant les tâches de manière efficace. Cela implique une définition claire des responsabilités de chaque membre même en étant réunis, ce qui nous permettra de maintenir une bonne dynamique. Vous trouverez ci-dessous une répartition provisoire des tâches (figure 2) .





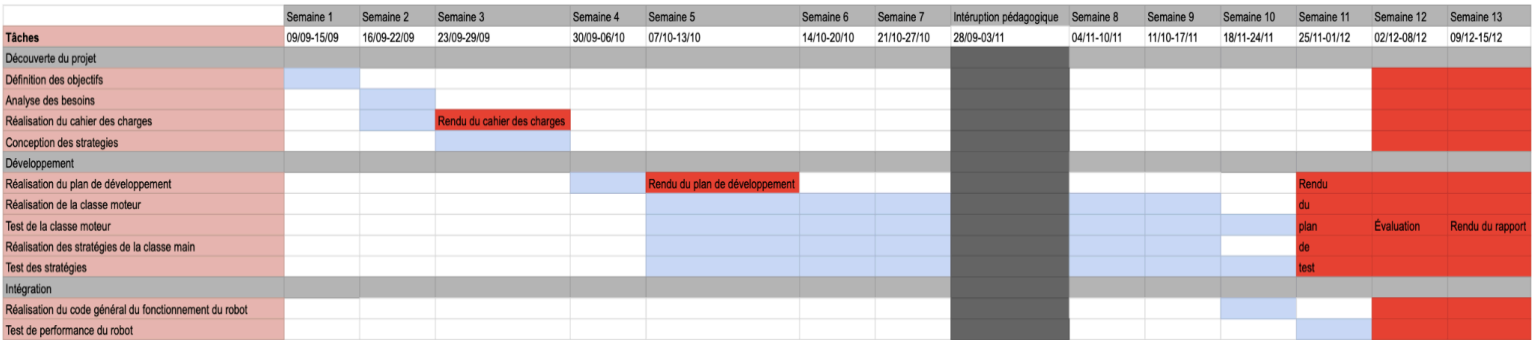
## 5) Glossaire

Termes	Définitions
Github	Logiciel de partage de code
Test de turing	Test visant à identifier pendant combien de temps une personne est capable de faire la différence entre une interaction avec une intelligence artificielle et un humain
Classe	En Java, une classe désigne la description d'un objet en termes d'attributs (ses données) et de méthodes (les opérations)
Object	En Java, un objet est une entité abstraite qui regroupe les données et opérations qui les manipulent
Ultrason	Onde sonore dont la fréquence est supérieure à 20 000 Hz et donc inaudible pour l'humain
Servomoteurs	Dispositif permettant de contrôler avec précision l'angle, la position ou la vitesse d'une pièce

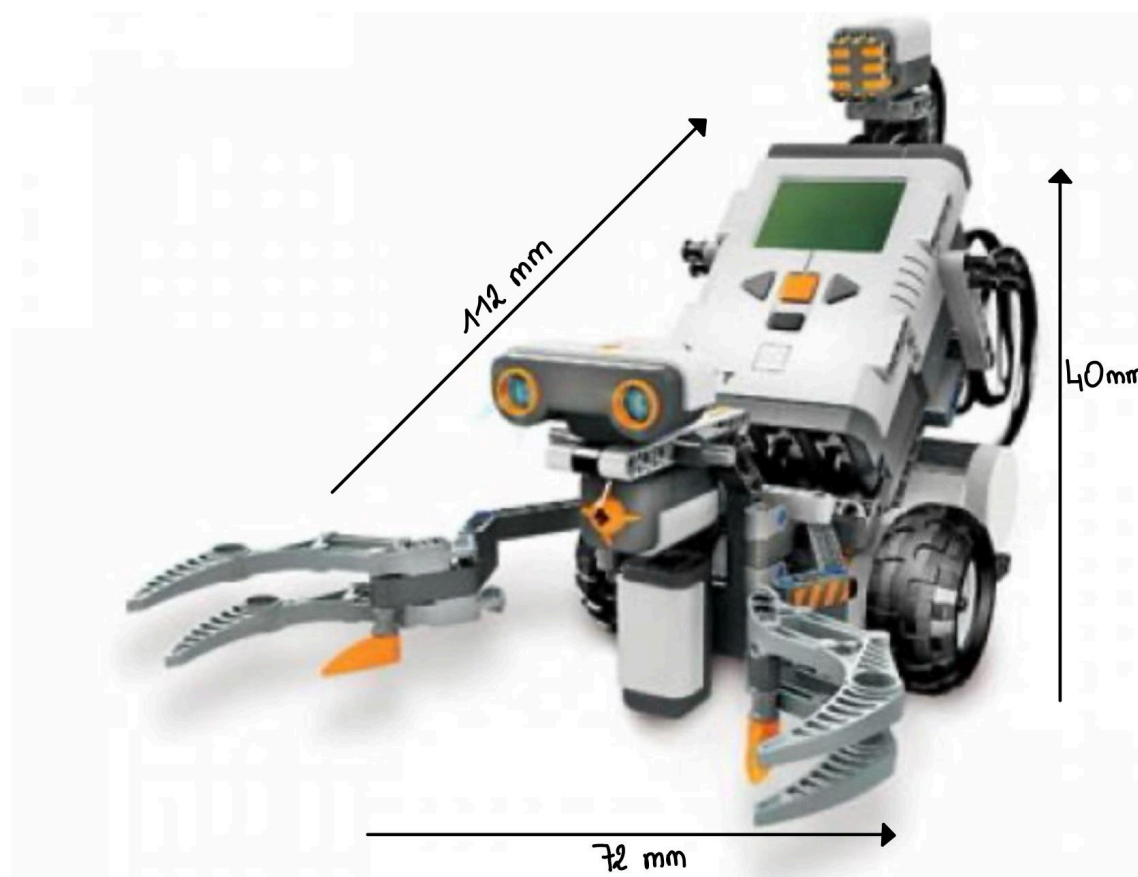


6) Annexes

Annexe I : Diagramme de Gantt



## Annexe II : Dimensions du robot



## 7) Références

- *Teaching : ia : project\_lego* [Damien Pellier Full Professor Univ. Grenoble Alpes]. (s. d.).  
[https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:ia:project\\_lego](https://lig-membres.imag.fr/PPerso/membres/pellier/doku.php?id=teaching:ia:project_lego)
- Pellier, D. (s. d.). *Introduction à l'Intelligence Artificiel*.  
[https://lig-membres.imag.fr/PPerso/membres/pellier/lib/exe/fetch.php?media=teaching:ia:01\\_introduction.pdf](https://lig-membres.imag.fr/PPerso/membres/pellier/lib/exe/fetch.php?media=teaching:ia:01_introduction.pdf)