

# CAHIER DES CHARGES



[https://fr.123rf.com/photo\\_110580265\\_bangkok-thailand-august-4-2018-robot-controls.html?vtl=07aeg4akz9luydjvtz-1-80](https://fr.123rf.com/photo_110580265_bangkok-thailand-august-4-2018-robot-controls.html?vtl=07aeg4akz9luydjvtz-1-80)

Initiation à l'intelligence artificielle  
2021/2022

Belaikous  
Dong  
Parouty  
Vallois

# Sommaire

<b>I-Introduction</b>	<b>1</b>
Historique et contexte	2
<b>II-Description de la demande</b>	<b>2</b>
Les objectifs	2
Produit du projet	2
Les fonctions du produit	3
<b>III-Contraintes</b>	<b>4</b>
Contraintes de délais	4
Contraintes matérielles	5
Autres contraintes	
<b>IV-Déroulement du projet</b>	<b>5</b>
Planification	5
Ressources	6
<b>V-Annexes</b>	
Figures	7

## **I-Introduction**

### **Contexte**

Dans le cadre du cours d'initiation à l'intelligence artificielle qui a lieu en L3, les concepts sont enseignés au cours de 24 heures de CM. Afin de se familiariser avec ceux-ci nous devons les opérationnaliser par la mise en œuvre d'un projet de groupe. Il s'agit de la programmation d'un robot dont le rôle est de ramasser des palets. Ce travail est évalué lors de la douzième semaine de cours lors d'un tournoi compétitif où l'objectif est pour les robots de chaque groupe de ramasser le plus de palets possible durant une période limitée. Une des stratégies consiste à récupérer le 1er palet avant l'adversaire puisque celui-ci rapporte plus de points.

### **Historique**

Le code est utilisé lors du tournoi mentionné dans l'introduction. Le robot programmé se retrouve face à un autre robot et le but à l'issue d'une manche de 5 minutes est d'avoir amassé le plus de palets possible et de les avoir déposés dans la zone d'en but adverse.

Au début du match, les robots sont positionnés sur la ligne délimitant leur zone d'en but respective (correspond aux emplacements R sur la figure 1) et font face à la zone d'en but adverse. Les palets sont disposés aux intersections entre les lignes de couleur (voir figure 1)

## **II-Description de la demande**

### **Les objectifs**

Il s'agit de créer les comportements d'un robot lui permettant de ramasser des palets et de les placer derrière une ligne blanche d'un terrain défini en prenant en compte les variables qui constituent son environnement. Le but, pour gagner, est d'obtenir le plus de points possible en récupérant le plus de palets.

Le robot obtenu est un agent autonome, rationnel et flexible. Il est capable de s'adapter à différentes situations de terrain, ce qui lui permet d'effectuer ce pourquoi il est programmé malgré les caractéristiques multiagent partiellement observable et non déterministes de celui-ci. Ses différentes fonctions et stratégies implémentées lui permettent de ramasser le plus de palets possible le temps du tournoi.

### **Produit du projet**

Le produit attendu est un code java fonctionnel permettant au robot d'atteindre les objectifs décrits plus haut. Le code se base sur l'API leJOS disponible sur le [site](#). Il regroupe plusieurs classes permettant de décrire les attributs et méthodes relatifs aux capteurs, moteurs et stratégies pour le comportement du robot.

Le code est remis ainsi que sa documentation interne sont déposés sur Github. Le client attend également des mises à jour régulières (hebdomadaire) afin de suivre l'avancée du projet.

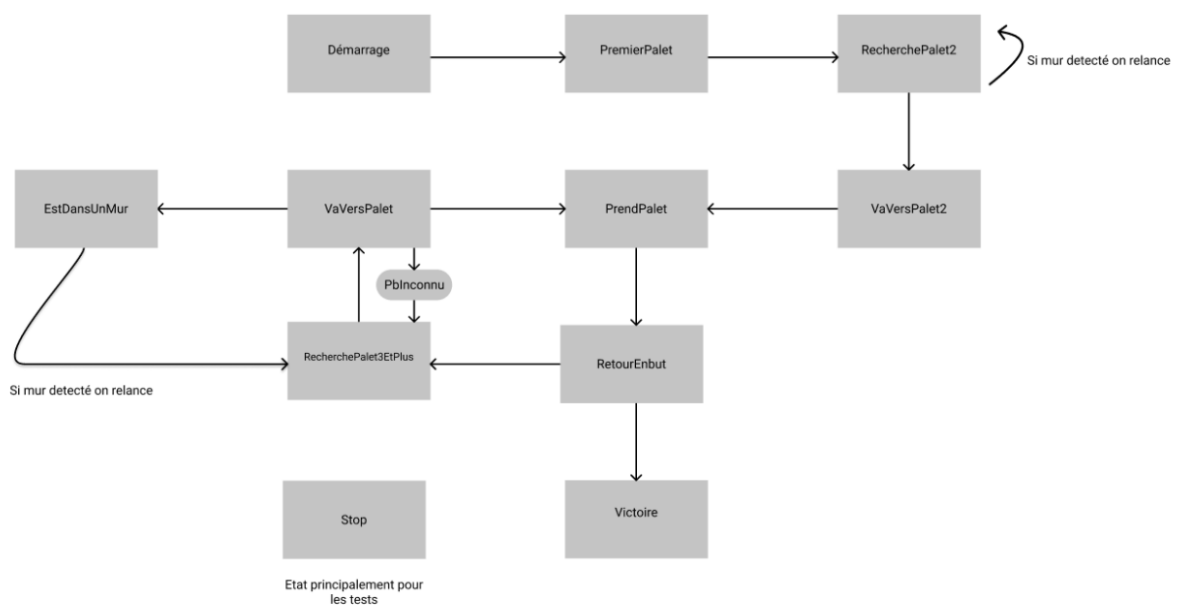
### **Les fonctions du produit**

Le code fourni doit permettre d'exploiter les effecteurs et capteurs de l'agent de sorte à ce qu'il ait les fonctions suivantes:

- Se déplacer dans l'environnement en utilisant un pilote pour coordonner l'action des deux roues.
- Faire appel à plusieurs fonctions et les exécuter simultanément.
- Au commencement du match faire appel à une stratégie spécifique au premier palet car l'environnement de départ est défini, il est donc possible d'exploiter ces informations pour économiser la recherche de palet.

- Identifier les murs par leur détection possible à moins de 30 cm du capteur et les éviter en tournant systématiquement à 90 degrés à droite puis relancer une recherche de palet si le robot n'a pas attrapé de palet ou utiliser la fonction lui permettant de se diriger vers la zone d'en but adverse s'il a attrapé un palet.
- Identifier l'arrivée à la zone d'embut adverse grâce au capteur de couleur qui permet de détecter la ligne blanche.
- Grâce à une boussole se tourner vers la zone d'embut adverse puis s'y diriger après avoir attrapé un palet.
- Grâce aux fonctions relatives au capteur de toucher, prendre connaissance d'un contact avec le palet et exploiter cette information pour refermer les pinces et attraper le palet.
- Détecter un palet attrapé si le capteur de toucher a été activé et les pinces fermées
- Prendre connaissance de l'action d'avoir déposé un palet puis se remettre en position de recherche.
- Grâce aux méthodes liées au capteur à ultrason et aux moteurs des roues, balayer un périmètre défini lors de la recherche de palets, identifier l'objet le plus proche et situé à plus de 30 cm du robot comme étant un palet puis se positionner en face de celui-ci.
- Grâce aux méthodes définies pour les moteurs et le capteur de toucher, se déplacer en ligne droite en ouvrant les pinces et s'arrêter une fois que le capteur de toucher rencontre un palet.
- Grâce au capteur de toucher et aux pinces, fermer les pinces suite à la prise de connaissance de la rencontre avec un palet.
- Après avoir fermé les pinces : se diriger vers la zone d'en but adverse si elles contiennent un palet, faire demi tour et recommencer la recherche sinon.

Pour schématiser l'utilité et le contexte de mise en œuvre de ces fonctions, le diagramme d'états suivant est développé.



### III-Contraintes

#### Contraintes de délais

Le produit doit être fonctionnel à la douzième semaine soit le 29 novembre 2021 qui correspond à la date du tournoi. Les livrables à rendre tout au long du projet sont :

- le cahier des charges à rendre la troisième semaine.
- le plan de développement à rendre la cinquième semaine
- le plan de test à rendre la dixième semaine
- Le code et la documentation interne à rendre la onzième semaine
- le rapport final à rendre la douzième semaine

Il faut respecter les échéances et réaliser le travail nécessaire dans la limite des 144 heures (TP inclus) de travail attribuées.

### **Contraintes matérielles**

Afin de rendre le produit demandé, nous avons besoin d'ordinateurs, de logiciels de programmation (eclipse), d'un accès aux documentations de gitHub. Nous avons aussi besoin d'accéder au robot fonctionnel (capteurs, moteurs, batterie etc..) et au terrain afin de tester notre programme.

### **Autres contraintes**

La compétition est régie par le [règlement de la compétition](#) qui impose certaines lois parmi les suivantes :

- Le robot doit passer les tests d'homologation avec succès(se déplacer, saisir un palet, déposer ce palet).
- La forme du robot est imposée et ne peut pas être modifiée.
- Le robot est autonome et ne peut pas être contrôlé à distance.
- Si le robot est perdu, le temps mort pouvant être demandé pour remédier au problème est limité à cinq minutes.
- L'accès au bâtiment Ampère/ terrain est limité.
- Il est impératif de maîtriser gitHub pour partager les documents liés au projet.
- Il faut se familiariser avec la programmation avec leJOS.

## **IV-Déroulement du projet**

### **Planification**

Pour la première phase qui est celle de la spécification, deux semaines sont consacrées à la conception du cahier des charges à partir de la semaine deux consacrée à la définition des objectifs, l'analyse des besoins et la planification. Il est rédigé et mis en page en semaine 3.

La seconde phase est celle de la mise en oeuvre du projet, elle s'étale de la première à la onzième semaines du projet et comporte trois axes qui sont :

- L'appropriation des fonctions de base qui dure de la première à la cinquième semaine, il s'agit de maîtriser les méthodes relatives aux capteurs et moteurs.

- Le développement qui s'étale de la deuxième à la dixième semaine. Il faut définir les différentes stratégies à implémenter à partir de la semaine deux, et rédiger un plan de développement à partir de celles-ci en semaine cinq. Ces stratégies sont implémentées testées et corrigées jusqu'à la dixième semaine.

- Les tests à effectuer sont définis à partir de la semaine deux et jusqu'à la semaine dix où la rédaction du plan de test est terminée. Les tests sont effectués jusqu'en semaine onze.

- Le rapport final est rédigé en semaine douze.

Pour la remise du travail, la configuration de gitHub s'étale sur les semaines deux et trois. Le code et sa documentation interne y sont implantés à partir de la semaine trois et jusqu'à la semaine onze. Les autres livrables y sont déposés aux échéances indiquées plus haut. Le tournoi a lieu en semaine douze.

## **Ressources**

En termes de ressources humaines, le groupe est composé de 4 personnes, nous avons également la possibilité de solliciter l'enseignant au cours du semestre.

Nous avons accès à nos ordinateurs, aux différents logiciels de programmation et documentations sur le sujet. Pour tester notre programme nous disposons du robot fourni par l'enseignant, son chargeur, sa batterie ainsi que du terrain où a lieu le tournoi.

La plateforme gitHUB est utilisée pour partager notre travail au sein du groupe mais aussi avec l'enseignant. Eclipse est utilisé pour produire le code et [le site de lejos](#) est utilisé comme documentation. Un logiciel de traitement de texte est utilisé pour produire les différents livrables.

## V- Annexes

### Figures

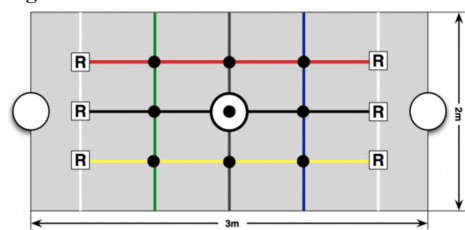


Figure 1: Schéma de principe du terrain