

# RAPPORT DE PROJET

***Chef de projet***

***Damien Pellier***

***Membres du groupe***

***Robin Chaussemy 12014629***

***Laurine De La Chapelle 12014762***

***Aurore Philippe 11900982***

***Mélisse Tilquin 12011231***

<b>1. Introduction</b>	<b>3</b>
<b>2. Guide de lecture</b>	<b>3</b>
<b>3. Travail de groupe</b>	<b>3</b>
3.1. Organisation	3
3.2. Points forts et difficultés rencontrées	3
<b>4. Conception</b>	<b>4</b>
4.1. Stratégie	4
4.1.a. Stratégie initiale	4
4.1.b. Difficultés rencontrées	4
4.1.c. Stratégie finale	4
4.2. Programmation	5
4.2.a. Code initial	5
4.2.b. Difficultés rencontrées	5
4.2.c. Code final	5
<b>5. Conclusion et discussion</b>	<b>6</b>
<b>6. Références</b>	<b>6</b>
<b>7. Glossaire</b>	<b>6</b>
<b>8. Index</b>	<b>7</b>

# 1. Introduction

Les programmeurs utilisant l'intelligence artificielle (IA) peuvent concevoir des algorithmes logiques, qui permettent à un agent, comme à un robot par exemple, de réaliser des tâches, qui imitent les tâches réalisées par des humains. Toutefois, cette imitation est souvent insatisfaisante et réductrice en comparaison de ce qui est réalisé via l'intelligence humaine.

Le projet présenté dans ce rapport est un projet en IA qui a pour objectif d'initier les étudiants le réalisant à l'IA en leur faisant programmer des robots via des algorithmes de l'IA.

## 2. Guide de lecture

Toute personne ayant des connaissances en robotique, informatique et IA peut supposément lire ce document sans se préoccuper du glossaire et des index.

Toute personne ayant aucune connaissance en robotique, informatique et IA, dans au moins l'un de ces domaines, et toute personnes ayant des difficultés dans au moins l'un de ces domaines doit supposément s'aider du glossaire, des index et/ou des références pour comprendre l'entièreté de ce document.

## 3. Travail de groupe

### 3.1. Organisation

Le travail de groupe à été répartie de manière à ce qu'il y ait deux groupes travaillant en simultanée ainsi que des sessions de travail en individuel. Les séances de cours du lundi ont servi de séance d'échange permettant des réunions de groupe. Nous avons fait le choix de répartir à chaque séance le travail à produire pour la séance suivante en fonction d'un échéancier préétabli.

Divers outils permettant de travailler à distance ont été utilisés, tels que l'application discord pour les échanges entre les membres du groupe, la plateforme GitHub pour les échanges des travaux réalisées par chacun des membres et des google docs pour réaliser les différents livrables.

### 3.2. Points forts et difficultés rencontrées

Les échanges au sein du groupe ont été difficiles, cela a entraîné des incompréhensions et des déséquilibres sur l'avancée du projet entre les membres du groupe. De plus, nous avons perdu la trace d'un membre sur les dernières semaines du projet ce qui a creusé un retard déjà existant sur l'avancée du projet. En effet nous n'avons pas réussi à récupérer une partie de notre travail qui se trouvait sur l'ordinateur du membre manquant, nous avons donc dû

recommencer à zéro cette partie du code ce qui a causé un retard conséquent dans l'échéancier prévu.

Les apports individuels de chacun des membres du groupe ont joué sur la qualité du projet et ont bénéficié au groupe. En effet, une personne avait des connaissances plus poussées que les autres en informatique et une autre avait plus de connaissances et de facilités en mathématiques que les autres. Cela a été avantageux d'une part mais cette différence inter-individuelle a aussi impacté négativement les échanges au sein du groupe et l'investissement de chacun sur l'ensemble des éléments à produire.

Pour que le groupe soit plus efficace, les échanges entre les membres du groupe aurait pu être plus réguliers et nombreux, les échanges avec le commanditaire aurait pu être plus nombreux, et l'investissement de chacun des membres aurait dû être plus important.

## 4. Conception

### 4.1. Stratégie

#### 4.1.a. Stratégie initiale

La première stratégie qui a été mise en place est basée sur l'utilisation de la caméra. Notre but était de produire un code permettant au robot de repérer l'ensemble des palets disponibles et de choisir via un calcul, réalisé à l'aide des données de la caméra et des capteurs du robot, le palet le plus optimal à récupérer et marquer. La caméra nous permettait d'avoir une vue d'ensemble du terrain et de ce qui s'y trouvait à tout instant. Une autre stratégie qui a été étudiée était de faire une recherche aux alentours du robot en réalisant des rotations d'un petit angle et d'effectuer une mesure de distance à chaque rotation cela permettrait de repérer la position des palets sur la table.

#### 4.1.b. Difficultés rencontrées

L'utilisation de la caméra n'a pas été un succès au vu de plusieurs de facteur telle la difficulté à connecter le robot et la caméra sur le même réseau, la calibration de la caméra à refaire à chaque séance, la transmission de trame qui n'est pas constante ce qui crée du délai dans la réception des données. Nous avons perdu beaucoup de temps en nous concentrant dessus. Nous avons dû revoir tardivement notre stratégie pour trouver une solution nous permettant de concourir sans l'utilisation de la caméra sans pour autant abandonner le projet de notre première stratégie.

#### 4.1.c. Stratégie finale

Après modifications, les phases d'homologations et la récupération du premier palet se font sans avoir recours aux données de la caméra. Les capteurs du robot sont les seuls à être utilisés. La récupération des autres palets se fait en ayant recours à la fois aux capteurs du robot et aux données de la caméra.

## 4.2. Programmation

### 4.2.a. Code initial

Dans un premier temps, nous nous sommes basés sur le premier diagramme de classes que nous avons construit. Il est possible de le trouver à la quatrième section de notre plan de développement. Nous avons décidé de découper le projet en trois grandes classes (Actionneur, Capteur, Agent) et une classe Stratégie qui permet de synchroniser tous les comportements du robot pendant la compétition. Mais suite au départ de notre camarade, nous avons redéfini la structure du projet en ajoutant les classes pour gérer la Caméra et une autre Outils\_Maths qui nous permet d'avoir des outils pour analyser les données de la caméra (norme, calcul d'angle optimal, calcul du coût de récupération de chaque palet grâce aux distances et l'angle séparant le robot et le palet). Cette évolution nous a amené à un deuxième diagramme de classes. Il est possible de le trouver à la cinquième section de notre plan de développement.

### 4.2.b. Difficultés rencontrées

Étant donné que les commandes leJos sont nouvelles pour nous, il a d'abord été compliqué de se familiariser avec les différentes classes que contient leJos. Il a aussi fallu qu'on prenne du temps pour se familiariser avec le robot. C'est-à-dire qu'il a fallu prendre du temps pour en comprendre le fonctionnement.

La connexion et les échanges entre le robot et la caméra a sûrement été ce qui nous a posé le plus de problème. En effet, il nous a fallu d'abord du temps pour comprendre comment procéder et comment coder les classes Camera, outils et stratégie. Nous avons aussi rencontré des problèmes techniques. En effet, La connexion du robot au réseau wifi sur lequel était connectée la caméra s'est souvent soldée par un échec, ce qui a grandement limité les test du code.

La calibration du robot nous a fait défaut. Nous avons passé du temps à essayé d'obtenir les valeurs du capteur de couleur du robot alors que les valeurs n'étaient pas fixes et peu fiables. De plus, le jour de la compétition, la reconnaissance du blanc par le robot n'était pas optimale, nous n'avons pas réussi à obtenir sa reconnaissance.

### 4.2.c. Code final

Finalement, nous avons opté pour une classe Actionneur, qui contient les ports moteurs ainsi que toutes les méthodes relatives aux actions que peut effectuer le robot, une classe Capteur, contenant les ports des capteurs de touché, d'ondes infra-rouge et de couleur du robot ainsi que toutes les méthodes qui utilisent et récupèrent les données des capteurs du robot, une classe Camera, contenant toutes les méthodes qui utilisent et récupèrent les données de la caméra, une classe stratégie contenant les méthodes relatives à la stratégie utilisée et une classe outils contenant les méthodes de calcul.

## 5. Conclusion et discussion

Le projet nous a permis d'avoir une vision appliquée de l'IA, d'améliorer nos savoirs-faire en programmation Java et d'acquérir de nouvelles compétences que ce soit en gestion de projet ou en informatique en général.

Au vu de la perte de temps dû aux difficultés rencontrées lors du projet, certains points spécifiés dans le cahier des charges ont dû évoluer comme notamment la fonction de perception de l'environnement du robot. L'échéancier prévisionnel du plan de développement a été respecté la plupart du temps sauf lors des dernières semaines où tout s'est un peu accumulé. La modélisation du code en UML qui était fixé n'a pas été totalement respectée car il y a eu une évolution dans la mise en place du code.

Au vu des difficultés rencontrées avec la caméra, nous aurions dû nous atteler à la réalisation d'une stratégie alternative qui n'utilise que les capteurs du robot durant toute la durée d'un match. Nous aurions été alors mieux préparés aux problèmes qu'il était possible de rencontrer le jour de la compétition.

Toutefois, notre groupe a su faire preuve de flexibilité et a essayé d'adapter le programme le jour de la compétition en fonction des problèmes rencontrés ce jour-là.

## 6. Références

Sources utilisées pour la rédaction du glossaire :

Robotique. (s. d.). Dans Larousse (Éd.), *Dictionnaire de français Larousse*.  
<https://www.larousse.fr/dictionnaires/francais/robotique/69649>

*Discord (logiciel)*. (2016). [https://fr.wikipedia.org/wiki/Discord\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Discord_(logiciel))

GitHub. (2009). <https://fr.wikipedia.org/wiki/GitHub>

*Google Docs : éditeur de documents en ligne*. (s. d.). Google Workspace.  
<https://www.google.fr/intl/fr/docs/about/>

## 7. Glossaire

Robotique : "Science et technique de la robotisation, de la conception et de la construction des robots."

Discord : "Logiciel propriétaire gratuit de VoIP et de messagerie instantanée."

GitHub : "Service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git."

Google docs : Application google qui permet de faire du traitement de texte en ligne.

## 8.Index

Projet (1. Introduction, p. 3)

Travail de groupe (3. Travail de groupe, p. 3)

Stratégie (4.1 Stratégie, p. 4)

Programmation (4.2 Programmation, p. 5)

LeJos (4.2.b. Difficultés rencontrées, p. 4)

Intelligence artificielle (1. Introduction, p. 3 ; 2. Guide de lecture, p. 3)

Robotique (2. Guide de lecture, p. 3)