

Plan de développement Intelligence Artificielle

Juillat Quentin

Cotard Adam

Mccarthy Brice

Thill Tao

Laucournet Tom

Table des matières

- Introduction 3
- Spécifications fonctionnelles 3
 - Description des fonctionnalités principales : 3
- Spécifications techniques 4
 - Langages et technologies utilisés : 4
- Organisation de l'équipe 4
- Développement et tests 4
 - Phases de développement 4
 - Stratégie de tests 4
- Annexe..... 5

1. Introduction

Par groupe de 5, nous devons programmer un robot EV3 en langage Java pour que lors de l'épreuve, ce dernier ramasse des palets les uns après les autres et les ramène au fur et à mesure dans notre camp. Cette épreuve se déroule dans un espace restreint, en compétition avec un autre robot placé sur le même plateau de jeu. Ce plateau contient 9 palets, chacun placé à une intersection de deux lignes de couleurs et notre objectif est de marquer plus de points que notre adversaire en sachant que le calcul des scores s'effectue comme cela :

- 3 points pour chaque palet placé dans la zone d'en-but adverse,
- 2 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.

2. Spécifications fonctionnelles

Cette partie décrit ce que le système du robot c'est-à-dire ce qu'il doit faire du point de vue fonctionnel, c'est-à-dire ses principales fonctionnalités et cas d'utilisation.

- **Description des fonctionnalités principales :**
 - Plusieurs stratégies ont été mises en place, mais la plupart du temps au lancement de l'épreuve, notre robot ramasse le palet situé en face de lui (ouverture des pinces, avance jusqu'à une pression du capteur de toucher à l'avant du robot, puis referme les pinces), et se dirige vers le camp adverse pour le déposer (se décale vers la droite de 45°, avance de 25 cm, tourne vers la gauche de 45° puis avance jusqu'à la ligne blanche du camp adverse).
 - Pour détecter un palet, notre robot recherche le palet le plus proche de lui en effectuant un tour sur lui-même et s'arrête dès qu'il repère un palet.
 - Ensuite, le robot se déplace en direction du palet observé, l'attrape et le ramène dans le camp adverse (s'oriente vers les cages adverses grâce à notre recalibrage puis s'y rend).

Ramassage des palets : Le robot doit identifier et ramasser les palets sur le plateau de jeu.

Détection des obstacles : Le robot doit détecter et éviter les obstacles (autres robots, murs, etc.).

Navigation : Le robot doit se déplacer dans le plateau et trouver les intersections et la zone d'en but.

Marquage de points : Le robot doit ramener les palets dans la zone d'en-but, c'est-à-dire une méthode pour s'orienter vers les buts, un autre pour s'y rendre et une dernière pour poser le palet

3. Spécifications techniques

Cette partie couvre les aspects techniques et l'architecture du projet.

- **Langages et technologies utilisés :**
 - Pour développer notre projet nous utilisons l'application Eclipse pour programmer en langage Java, de plus l'outil GitHub est essentiel pour centraliser notre projet et apporter des modifications à notre code. Enfin, la bibliothèque numérique en ligne de LeJOS est une ressource essentielle pour notre projet.

4. Organisation de l'équipe

Pour cela, un [échéancier](#) a été créé (annexe), ainsi qu'un GitHub

- GitHub : <https://github.com/AdamInLoveOfBrice/miashsia/>

5. Développement et tests

- **Phases de développement**

Notre code est organisé de la façon suivante :

Package Capteur :

- Classe CapteurDistance :
- Classe CapteurCouleur :
- Classe CapteurToucher :

Package Moteur :

- Classe Pince :
- Classe Roue :

Package IAPackage :

- Classe Principal :

- **Stratégie de tests**
 - Tests unitaires sur chaque module (capteur de distance, couleur, pince, etc.).
 - Tests en conditions réelles (plateau de jeu, ou sur le sol) pour ajuster les comportements, les angles, les diamètres des roues, et d'autre problème.

Annexe : Echéancier IA

