

RAPPORT DU PROJET INTELLIGENCE ARTIFICIELLE PROJET YETI



Sommaire :

I. Tactique

1. Exécution du programme
2. Système GPS
3. Détecter et esquiver les obstacles

II. Architecture du code

1. Description des packages
2. Diagramme UML

I. Tactique

Sachant que le plateau dispose de neuf palets qui sont aux croisements de chaque ligne. Le robot doit marquer des points en amenant les palets dans le camp adverse rapidement tout en évitant les obstacles (mur c'est-à-dire bordure du terrain, adversaire).

Notre robot dispose d'un système GPS qui lui permet de savoir où il se situe exactement sur la plateau, notamment il est capable de retrouver son chemin à partir de n'importe où sur le plateau. Il dispose aussi des stratégies permettant de réagir aux imprévus.

I. 1- Exécution du programme

Dans cette partie on va parler des étapes d'exécution du programme c'est-à-dire l'ordre que le robot suit pour récupérer les palets.

Avant de lancer notre robot, on le place où on veut et puis on choisit les deux couleurs correspondantes à l'emplacement du premier palet qui se trouve devant lui (voir annexe Image 1).

- I. **Premier palet** : Le robot est placé au début de la ligne choisie. Il avance tout droit à vitesse moyenne , quand il détecte le palet il ferme les pinces puis se met en diagonale et avance au milieu de la case (carrée), se remet en position puis avance tout droit jusqu'à ce qu'il rencontre la ligne blanche, là il ouvre ses pinces (dépose le palet) et recule laissant ainsi le palet dans sa zone de but. Dans cette étape le suivi de ligne n'a pas été utilisé pour gagner du temps.
- II. **deuxième palet** : ensuite le robot se tourne vers la ligne (bleu ou verte), il cherche le palet le plus proche (sans suivre de ligne), il détecte le palet, ferme ses pinces puis il fait demi-tour et replace le palet comme précédemment. Notre robot connaît parfaitement les emplacements des palets sur le plateau de jeu. Si il ne détecte pas de palet à son emplacement, il s'arrête puis il va chercher le palet le plus proche encore une fois et ainsi de suite ..
- III. **Troisième palet** : Même stratégie que le deuxième palet, on dépose le palet, recule, puis chercher le palet le plus proche encore une fois.
- IV. **Les autres palets** : Le robot va refaire la même tactique pour tous les palets jusqu'à ce qu'il y a plus de palets sur le plateau.

II. 2- Système GPS

Le système "GPS" utilise les tachymètres intégrés dans les roues du robot pour mesurer précisément la distance parcourue par chaque roue, et donc en déduire la position et la rotation approximative du robot. Cette position est mise à jour dès qu'une ligne (de position connue) est rencontrée.

III. 3- Détecter et esquiver les obstacles

Pour esquiver les obstacles on a deux tactiques différentes :

- ➔ Si la distance entre notre robot et l'adversaire est inférieur ou égale à 10cm :

On détecte l'adversaire, on fait un demi-tour puis on cherche la ligne noir, pour ensuite à suivre pour récupérer les palets.

- ➔ Si la distance entre notre robot et l'adversaire est entre 10 et 30 cm :

On détecte l'adversaire, on tourne à droite, puis on avance pour le dépasser et après on tourne à gauche pour revenir sur la ligne de départ.

II. Architecture du code

1. Description des packages

Notre code se compose de six packages :

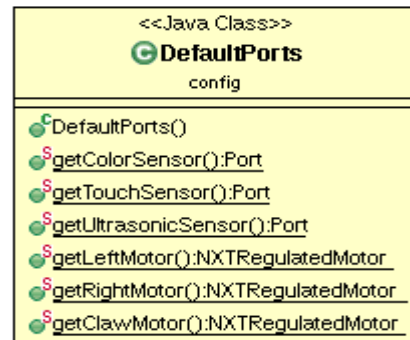
- Package config : constitué d'une seule class qui permet de configurer les ports et moteurs du robot.
- Package gps: constitué de 5 class qui permettent de faire fonctionner le système gps décrit dans la première partie.
- Package main : avec la class du main où on peut compiler notre programme.
- Package motor : avec deux class qui gèrent les méthodes de mouvements des pinces et des roues.
- Package sensors : avec 6 class qui permettent de faire fonctionner les capteurs (de couleur, d'ultrason)
- package Strategy : constitué de 12 class qui permettent de gérer les stratégie utiliser par notre robot.

2. Diagramme UML

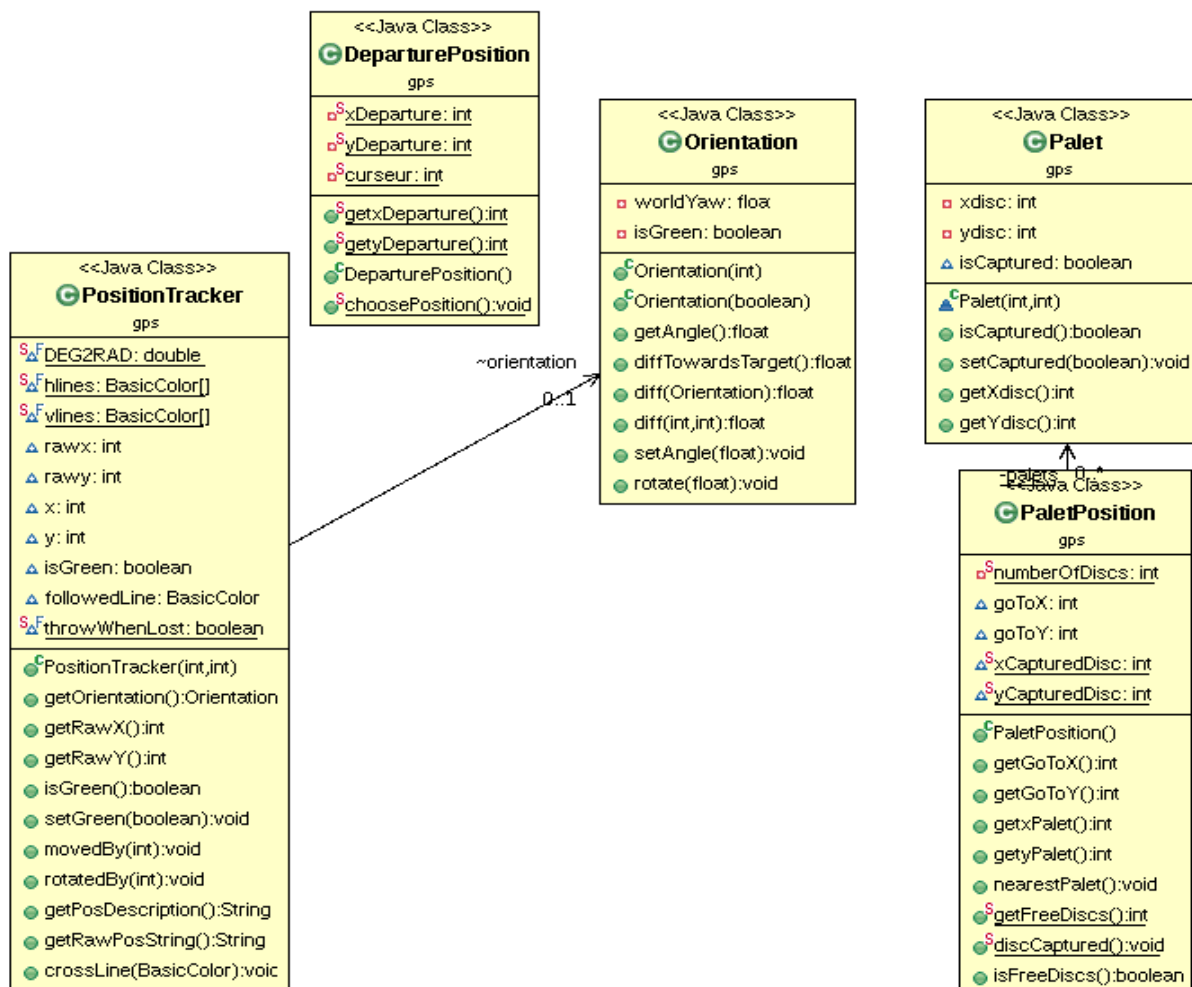
Des diagrammes UML sont fournis pour chaque package Java en format image PNG et au format UCLS. Ils se trouvent à la racine du dossier rendu en haute qualité.

Voici un aperçu de ces derniers :

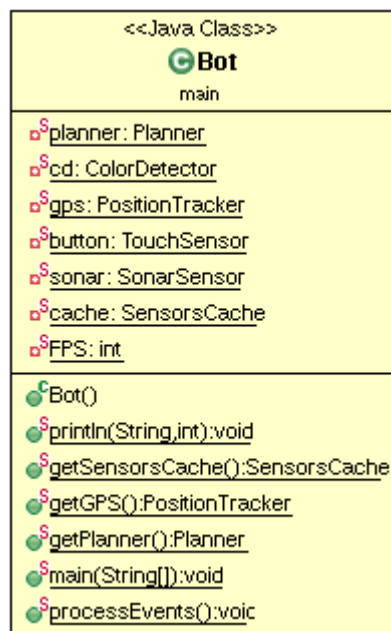
Package config :



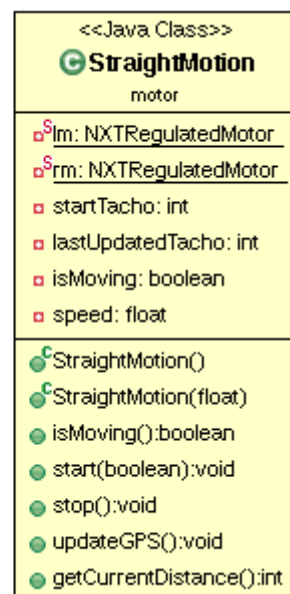
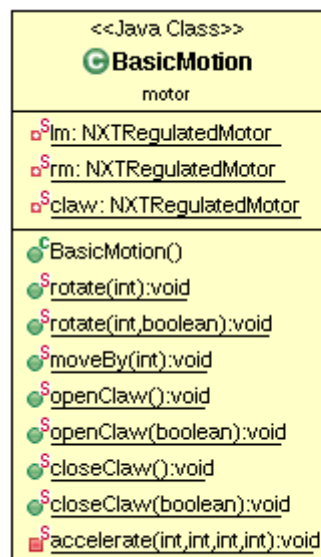
Package gps :



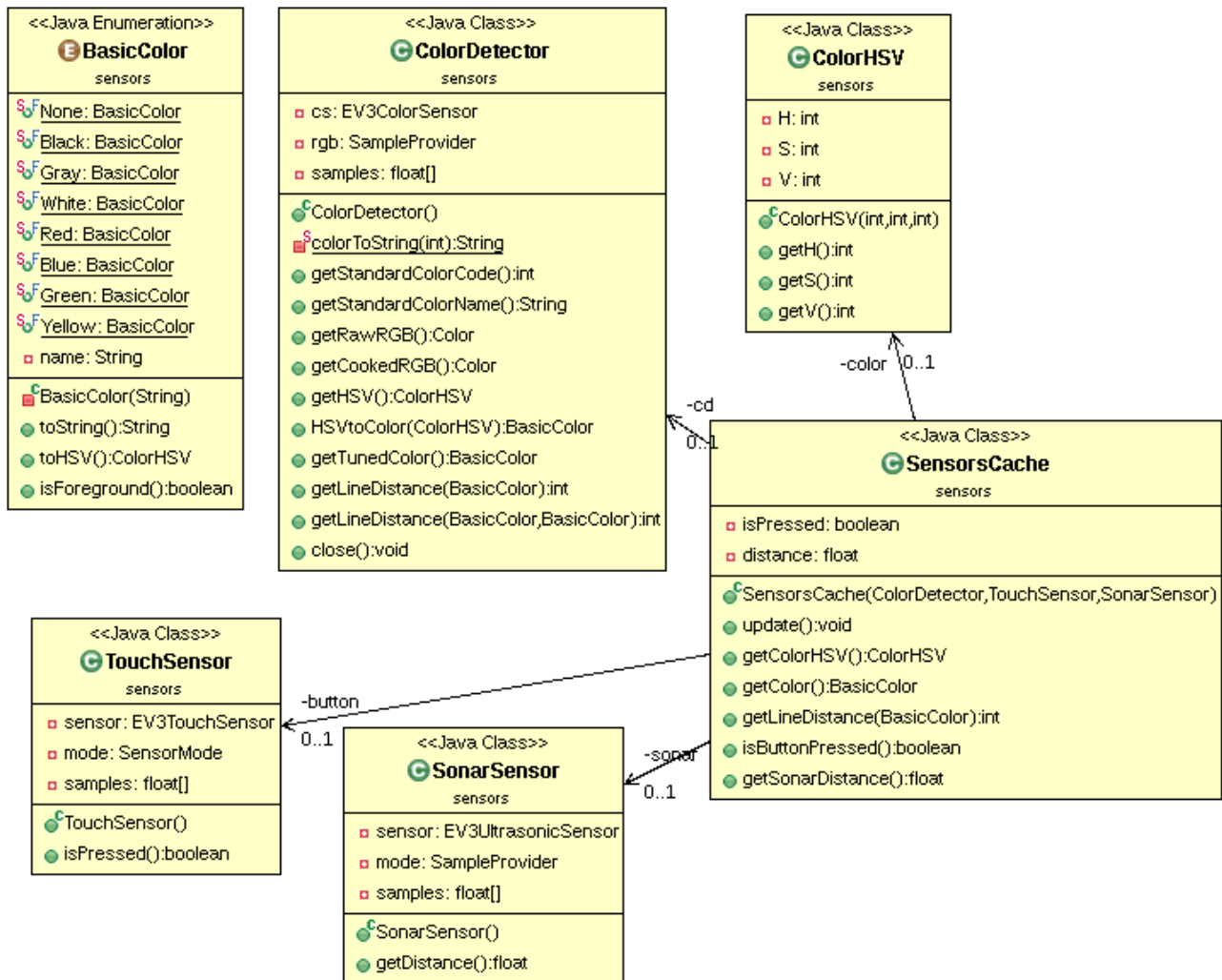
Package main :



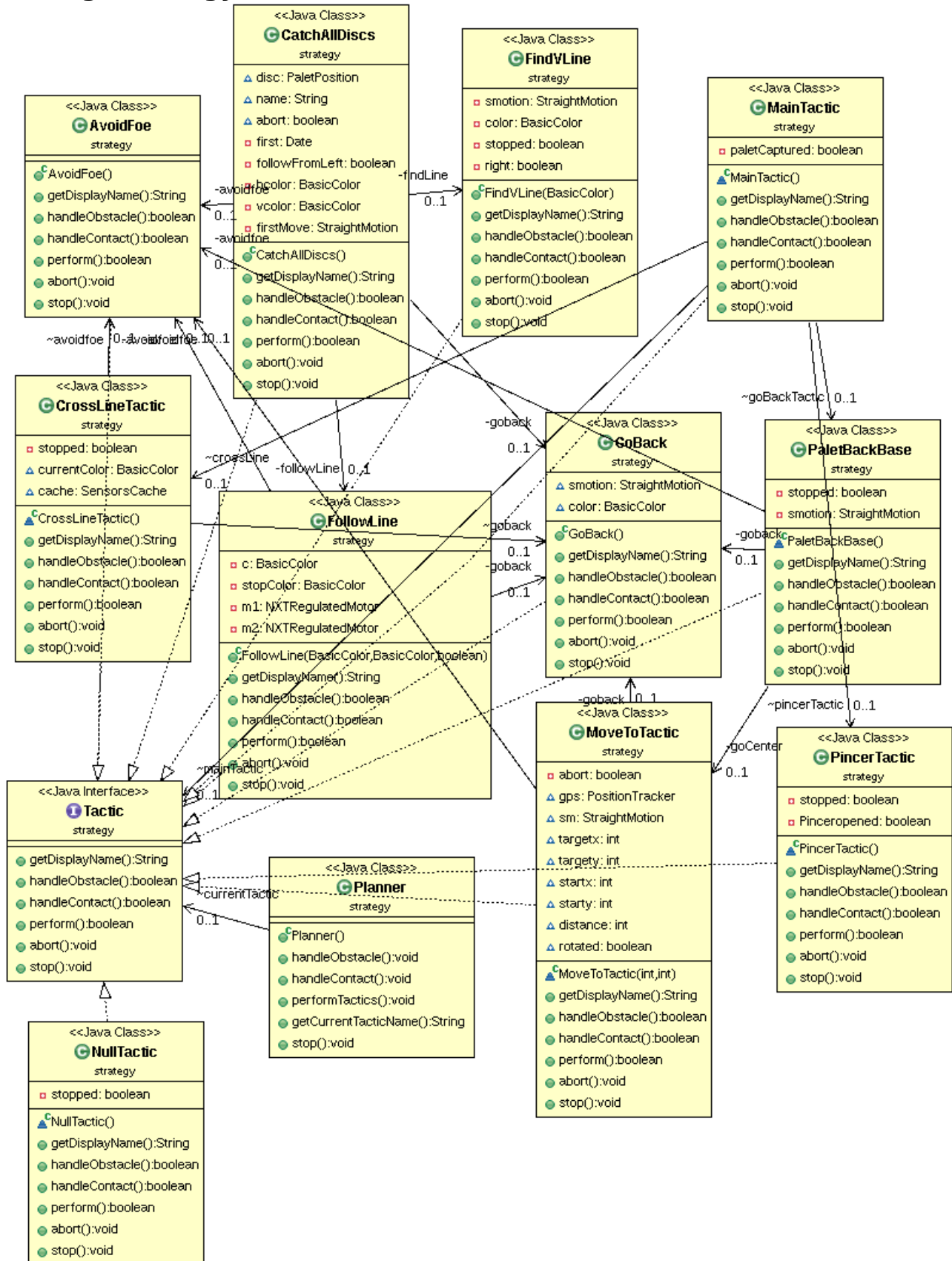
Package motor :



Package sensors :



Package strategy :



Annexe :

