

PLAN DE DÉVELOPPEMENT

ROBOT LEJOS EV3

VERSION DU DOCUMENT : 1.0

NOM DU PROJET : [ROBOT-LEJOS-EV3-FASTLEARNING](https://github.com/Vetrarr/-Robot-Lejos-EV3-FastLearning)

DATE : 09/10/2022

AUTEURS : LOUIS BASTIEN FAUCHON

DANIEL BEQAJ

RAPHAËL DELATTRE

TYPE DE DIFFUSION : [HTTPS://GITHUB.COM/VETRARR/-ROBOT-LEJOS-EV3-FASTLEARNING](https://github.com/Vetrarr/-Robot-Lejos-EV3-FastLearning)

MAÎTRE D'OUVRAGE : DAMIEN PELLIER

MAÎTRE D'OEUVRE : VOIR AUTEURS

Introduction

1.1 Objet

1.3 Documents de référence

2 Organisation du projet

3 Méthode de développement et calendrier directeur

4 Plans d'itérations

4.x Plan pour l'itération x

4.x.1 Tâches et calendrier

4.x.2 Contenu des tâches

4.x.3 Produits réalisés

5 Maîtrise des risques

6 Méthodes, techniques et outils

7 Infrastructures et moyens

Introduction

1.1 Objet

Une compétition de robots Lejos EV3 sera organisée prochainement et les robots doivent fonctionner avec une programmation embarquée, ce document vise à clarifier l'organisation de l'équipe, la méthode de développement et le calendrier d'avancement.

1.2 Terminologie et sigles utilisés

JDK = Java Development Kit

Itération = étapes à répéter

1.3 Documents de référence

Le cahier des charges :

https://github.com/Vetrarr/-Robot-Lejos-EV3-FastLearning/blob/main/Cahier%20Des%20Charges_v1.pdf.

2 Organisation du projet

L'équipe est composée de 3 membres, tous étudiants en 3ème année de licence MIAHS.

L'organisation de l'événement et la mise à disposition du matériel est assuré par l'enseignant Damien Pellier.

Chaque membre est responsable du robot et de l'avancée de la programmation.

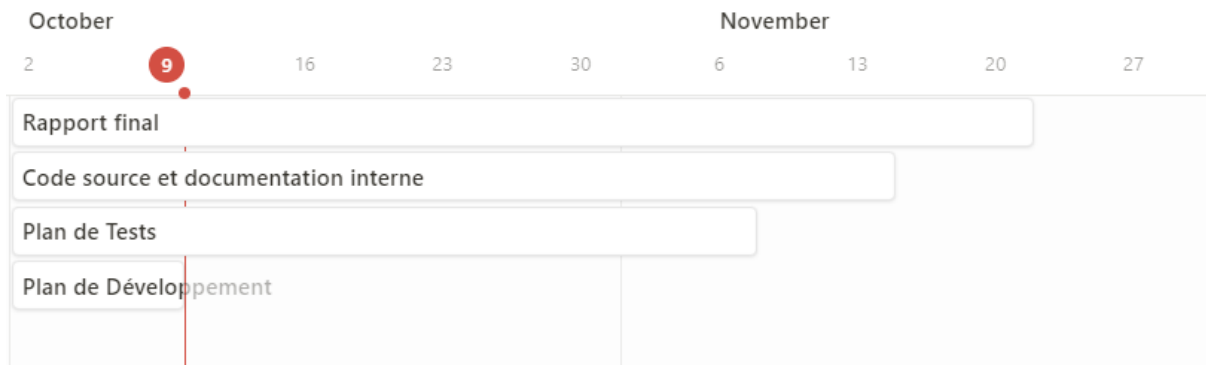
3 Méthode de développement et calendrier directeur

Les étapes initiales sont la découverte du robot, de ses capteurs et actuateurs, et aussi de l'environnement de la compétition. Dans un premier temps des mesures sont effectuées sur chaque capteur afin de déterminer le type des données, leur fréquence d'échantillonnage et leur utilisation qui est fonction des actions du robot. Des tests sur ses différents actuateurs sont aussi mis en place afin de vérifier que tout fonctionne.

Dans un deuxième temps, les performances du robot sont déterminées. Les performances du robot dépendent des phases de la compétition.

Le robot doit pouvoir se rendre dans l'en-but adverse depuis le point de départ. Le robot doit pouvoir se saisir d'un palet au centre du terrain et le déposer dans la zone d'en-but. Le robot doit placer le premier palet dans l'en-but adverse le plus rapidement possible, le robot doit placer le maximum de palet (9) dans l'en-but adverse et conserver un palet entre ses pinces à la fin des 5 minutes.

Les tests qui seront réalisés sur le programme du robot et sur le robot directement doivent être prévus d'ici la semaine du 7 novembre, la documentation pour la semaine du 14 novembre, et un rapport de l'ensemble du projet la semaine du 21 novembre.



4 Plans d'itérations

Les tâches à réaliser sont les suivantes :

Proposer des indicateurs de performance pour les différents objectifs de la compétition des robots. Le programme du robot est organisé en 3 classes, senseurs, actuateurs et performance.

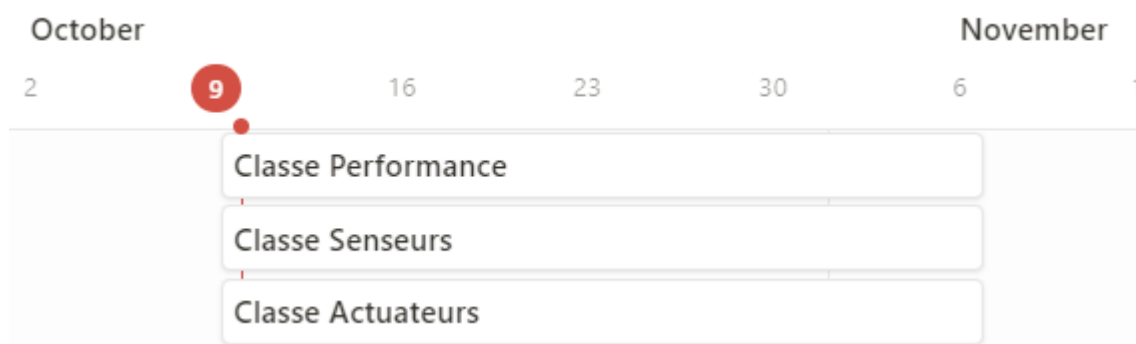
La classe senseur doit permettre de détecter l'environnement, les murs, les palets, à distance et entre les pinces, l'autre robot.

La classe actuateurs doit permettre d'avancer dans une direction, de faire demi-tour, de se replacer dans une direction (un palet, l'enbut ...), d'ouvrir et fermer les pinces.

La classe performance doit permettre d'utiliser le temps minimal pour placer les palets dans l'en-but adverse, de trouver le palet le plus proche, le trajet le plus rapide depuis sa position et une position cible.

Le programme sera régulièrement mis à jour pour répondre à un mauvais fonctionnement ou à une optimisation.

Ces trois classes seront développées et mises à jour pendant les 4 semaines du 10 octobre au 6 novembre.



4.x Plan pour l'itération x

4.x.1 Tâches et calendrier

Un point de départ pour commencer à coder les classes est de faire des programmes test pour essayer d'intégrer différentes fonctionnalités du robot avec un objectif précis.

Par exemple en testant un programme "attraperPalet" qui avait pour objectif d'avancer vers un palet devant, l'attraper et faire un demi tour pour le poser à la fin, on s'est rendu compte:

- des défauts de notre programme (quand le capteur envoie des valeurs qu'on avait pas anticipé)
- comment l'améliorer (faire des nouveaux cas pour les valeurs INFINITY)
- quoi factoriser comme code dans les trois classes importantes (la partie du programme qui fait avancer le robot tout droit pourrait être dans une méthode de la classe actuateurs)
- le prochain programme à tester (exemple le robot est capable d'attraper même un palet situé dans d'autres endroits).

Ce plan est répété après avec le prochain programme test, donc la prochaine itération, jusqu'au moment où les trois classes principales sont complètes et le code est factorisé au maximum pour pouvoir être réutilisé dans le futur quand on en aura le besoin.

Les tâches correspondent aux différentes méthodes listées ci-dessous :

CLASSE ACTUATEURS{

Attributs

Constructeurs

Méthodes

AvanceSynchronise() //LigneDroite

AvanceAsynchrone() //Virage

RoueGauche()

RoueDroite()

LigneDroite(distance, vitesse)

VirageSurPlace(angle)

VirageEnAvant(angle,distance)

OuverturePince()

FermeturePince()

A

}

CLASSE SENSEURS{

Attributs

Constructeurs

Méthodes

Ultrason() : retourne la valeur de l'ultrason

Toucher() : retourne un booléen, palet ou pas de palet

Couleur() : valeur du RED

Position() : retourne la position de la caméra infrarouge

Vitesse() : retourne la vitesse des servomoteurs

}

CLASSE PERFORMANCE[

Attributs

Constructeurs

Programme

Méthodes

```
DetectionPalet()  
PaletLePlusProche()  
RepositionPaletLePlusProche()  
SaisiePalet()  
RepositionVersEnBut()  
DepotPalet()  
}
```

4.x.2 Contenu des tâches

Chacun des membres de l'équipe est responsable du bon fonctionnement du robot et de son utilisation. Chacun met à jour la programmation et est capable d'utiliser le code des autres membres de l'équipe pour améliorer le fonctionnement du robot. Pour tout ajout de code, il doit être prévu un test unitaire, un test global et un test en fonctionnement réel. Des commentaires sont prévus pour la documentation. Et un court bilan sur la semaine passée sera réalisé chaque semaine pour le rapport final.

4.x.3 Produits réalisés

Un cahier des charges qui spécifie les besoins, contraintes et objectifs.
Un plan de développement (ce document).
Un plan de test pour valider toutes les réalisations précédentes.
La documentation interne du code et le code source.
Le rapport du Projet.

Le(s) programme(s) du robot.

5 Maîtrise des risques

Les risques sont la destruction partielle du robot, pour l'éviter, le robot est transporté avec soin. La non réalisation des tâches du robot, pour éviter cela des performances minimales sont prévues, la mise à jour du code se fait en conservant l'historique afin de pouvoir retrouver un fonctionnement plus simple et performant.

6 Méthodes, techniques et outils

Lister les standards dont l'application est demandée pour la spécification, la conception, le codage...
Indiquer les outils nécessaires pour réaliser le projet.

Les outils utilisés pour la réalisation du projet sont

-l'IDE eclipse

-lesWiki et API disponible ici : <https://lejos.sourceforge.io/ev3.php>

-un réseau wifi.

7 Infrastructures et moyens

Faire une synthèse des infrastructures et moyens nécessaires pour réaliser le projet : matériels, outils logiciels, réseaux, bâtiments...

Préciser la disponibilité de ces infrastructures et moyens, et, pour les infrastructures ou autres moyens à acheter ou développer, les travaux à réaliser et leur calendrier.

La plupart des outils logiciels dont on a besoin pendant ce projet sont gratuit à télécharger et open source. Les plus importants sont:

- Eclipse (disponible sur www.eclipse.org)
- JDK 1.7 (disponible sur www.oracle.com)
- librairie lejos (trouvé sur www.lejos.sourceforge.io)
- le robot EV3 (fourni au départ du projet)