

Robot suiveur de ligne


Année : 2018/2019

Equipe :

- Djebbar Wail
- Thaï Kévin
- Zniber Mohammed Abbas



Sommaire

- 
- Introduction
 - Présentation des fonctionnalités
 - Architecture, conception et gestion du projet
 - Code
 - Déroulement du projet
 - Conclusion


Introduction

- Notre robot suiveur de ligne est un ensemble composé d'une brique ev3, capteur ev3 color sensor, deux moteurs et des pièces de lego. Ce dernier a une grande importance dans le processus de fabrication industrielle, l'automatisation ainsi que le transport dans une direction spécifique (bus ou métro sans chauffeur).
- L'importance de notre projet se situe dans la recherche de l'efficacité et l'optimisation d'un programme, permettant un suivi de ligne correcte dans un meilleur délais sur tout type de circuit.





Présentation des fonctionnalités

- Apprentissage de couleurs
 - Reconnaissance de couleurs
 - Suivi de ligne : Balayage de gauche à droite
 - Suivi de ligne : Balayage de gauche à droite avec traversée
 - Suivi de ligne : PID
- 



Quels sont les scénarios d'utilisation principaux ?

- Scénario basé sur la vitesse.
- Scénario basé sur la précision.
- Scénario basé sur l'équilibre entre précision et vitesse.



Architecture, conception et gestion du projet

- Décomposition des problèmes du logiciel :
 - Un problème = une classe
 - Un sous-problème = une fonction
 - Compétences techniques : Java, Conception POO
 - Découpage : Couplage faible, forte cohésion



Architecture, conception et gestion du projet

- Répartition des tâches et conception :
 - Montage robot (3)
 - L'apprentissage et la reconnaissance de couleur (3)
 - Suivi de ligne : algorithme naïf (2)
 - Suivi de ligne (avancée) (2)
 - Pid (2)
- Classes principales:
 - Apprentissage.java – reconnaissance.java – SuivreLigneBalayage.java – SuivreLigneBalyageTraversee.java - PID.java
- Point fort de l'architecture du code :
 - réutilisabilité, maintenabilité, adaptation



Architecture, conception et gestion du projet

- Difficultés rencontrées :

- Gestion de la vitesse

- Calcul de la bonne couleur apprise

- Précision

- Solutions :

- Ajustement de vitesse avant le lancement du programme

- Calcul, après plusieurs scans

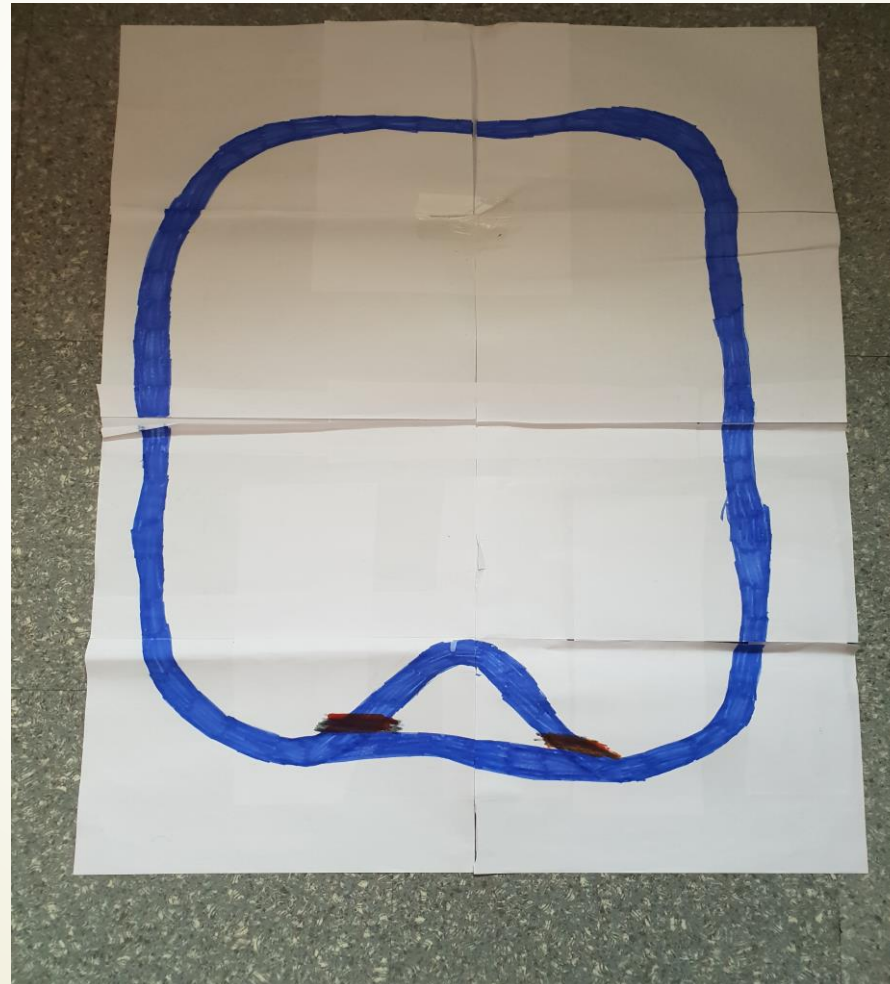
- Réduction de vitesse

- Testes :

- Tapis personnel fabriqué pour tester chez soi (Image)

- Les 3 tapis à disposition dans la salle de TD.

Tapis personnel



Code

```
double t=spd1*2.5;
spd2=(int)t;
EV3ColorSensor colorSensor = new EV3ColorSensor(SensorPort.S3);
EV3 ev3 = (EV3) BrickFinder.getLocal();
Reconnaissance r = new Reconnaissance(colorSensor);
leftMotor.setSpeed(spd1);
rightMotor.setSpeed(spd2);
leftMotor.forward();
rightMotor.forward();
int direction=0;
int accrocher=0;
int trav = 0;
while(Button.DOWN.isUp()) {
    if(r.aff()!=0) { // si la couleur est différente de la ligne
        if(accrocher == 1) { //si il a trouvé la ligne
            if(trav == 1) { // si il a traversé la ligne
                trav =0 ;
                if(direction == 0) direction = 1 ; // changement de direction
                else direction = 0 ;
                inverser(leftMotor,rightMotor,direction,spd1,spd2); //inverser la vitesse des moteurs
            }
        }else { //si il n'a pas trouvé la ligne
            leftMotor.setSpeed(spd1);
            rightMotor.setSpeed(spd2);
        }
    }

    }else { //si la couleur est la couleur de la ligne
        if(trav ==0) { //si il n'a pas encore traversé
            trav = 1;
            accrocher=1;
        }
    }
}
```



Déroulement du projet



- Octobre : Reformatage, installation du firmware SDK Mindstorm (+ Java 7 Max), plugin LeJOS pour Eclipse, Premier programme et fonctionnement des moteurs.
- Novembre : Première version de l'implémentation du programme d'apprentissage et de reconnaissance de couleurs.
- Décembre : Montage de la brique, suivi de ligne droite et première version de suivi de ligne courbée
- Janvier : Amélioration de l'algorithme d'apprentissage et de reconnaissance ainsi que de l'algorithme de suivi de ligne (1min7sec).
- Février : Optimisation du suivi de ligne avec balayage (20sec).
- Mars/Avril : Implémentation du suivi de ligne PID et l'algorithme suivi de ligne balayage avec traversée (16sec).



Conclusion

- Les acquis
programmation embarqué, robotique.
- Version II logiciel
apprentissage du circuit, dessiner le parcours, régler la vitesse aux virages,
l'ajout gyro sensor ev3.
- Les imprévus
capteur défectueux (nxt), manque de précision pour le capteur (ev3),
manque de pièce, prendre en compte la distance entre le capteur et le sol.
- Solutions
réalisation de plusieurs scans, changer de capteur, bien fixer le capteur.