



Projet long: Mindstorm

Johann Hospice
Naïda Meftah

Introduction

Robot suiveur de ligne

Intérêts principaux :

- ludique : tournois, jeux...
- industriel : assistance à la production automatisée
- logistique : transports automatiques

Problématiques :

- suivi précis de la ligne
- rapidité du robot



Comment permettre au robot de suivre une ligne précisément le plus rapidement possible ?

Fonctionnalités

Apprentissage de couleurs

Enregistrement de plusieurs couleurs

Reconnaissance de couleurs

Distinction du robot entre les couleurs ajoutées

Suivi de couleur sur une ligne

Plusieurs méthodes (algorithmes) de suivi de la ligne

Modification rapide de la vitesse du robot

Utilisation des boutons pour régler rapidement la vitesse du robot

Architecture, conception et gestion de projet

Outils utilisés

Git, Java, Eclipse, IntelliJ IDEA

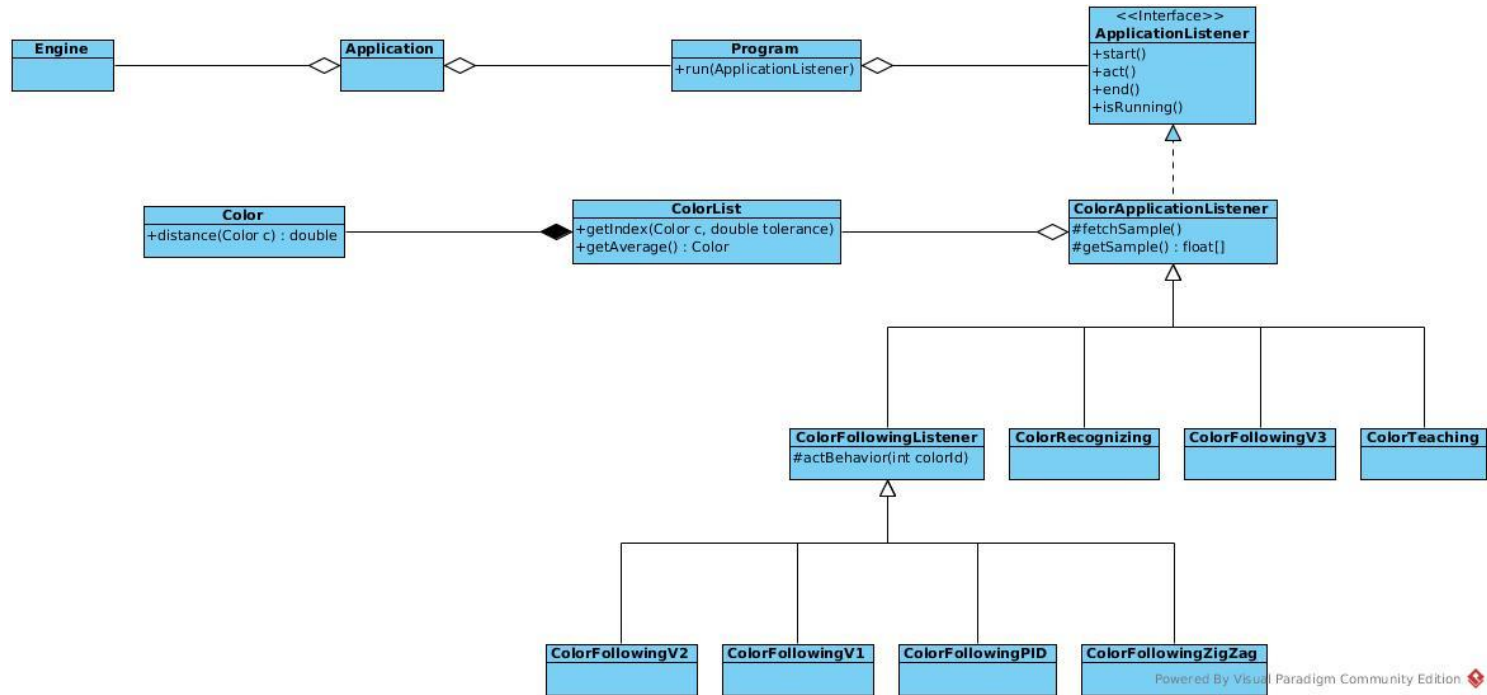
Compétences techniques requises

Programmation Java/Lejos

Décomposition en sous-problèmes :

- enregistrer/reconnaître des couleurs
- suivre une ligne
- optimiser l'algorithme

Paquets et dépendances : Lejos EV3



Conception orientée objet : classes et leurs relations

Rôles de chacun des membres et temps de développement

Tests du projet

Test sur tapis de ligne

Difficultés technologiques/techniques :

- installation de l'environnement (robot)
- compatibilité à Windows
- utilisation de la brick et ses modules
- test en dehors de l'université
- algorithme PID

Programmation

```
// K pourcentage de rotation, SPEED vitesse maximale
private static final int SPEED = 600, K = 45;

// rightSide dit sur de coté on se trouve sur la ligne, changedDirection dit si on a changé de direction
private boolean rightSide = true, changedDirection = false;

/**
 * @param colorId index de la couleur reconnue
 */
@Override
protected void actBehavior(int colorId) {
    switch (colorId) {
        case 0: // si sur la ligne, tourne à droite ou à gauche
            if (rightSide) {
                goLeft();
                changedDirection = false;
            } else {
                goRight();
                changedDirection = false;
            }
            break;
        case 1: // si sur le fond on change de direction
            if (!changedDirection) {
                rightSide = !rightSide;
                changedDirection = true;
            }
            break;
    }
}
```

Algorithme de suivi de ligne “Zig Zag”

Conclusion

Une expérience nouvelle

Amélioration des paramètres de vitesse,

Approfondir l'algorithme PID

Programmer différents comportements selon la couleur reçu