
Robot Suiveur de Ligne

— Lahdir Mohamed Akram —
Khengui Salim

Introduction

Qui somme-nous ?

Bonjour, Nous somme Khengui Salim et Lahdir Mohamed Akram, des étudiants en M1 informatique parcours IMPAIRS et DATA respectivement à l'Université Paris Diderot.

Quel est l'objectif du projet ?

L'objectif de ce projet et de développer un robot suiveur de ligne, qui peut gagner la compétition des suiveur de ligne à la fin de semestre.

Introduction

Que fait le robot ?

Le robot suiveur de ligne, comme son nom l'indique, est un véhicule automatisé, qui suit une ligne visuelle intégrée au sol.

Quels sont les principaux scénarios d'utilisation ?

Il y'en a plusieurs cas d'utilisation des suiveurs de ligne, certains d'entre eux sont: tournois de suiveurs de ligne, assistance dans les processus de production automatisé, services de livraison, transport en commun ... etc.

Architecture, conception et gestion de projet

Pour faire ce projet, nous avons divisé notre travail en jalons, chacun avec un petit problème à résoudre pour résoudre le grand problème ce qui est, développer un robot suiveur de ligne. Les jalons qu'on a fait sont :

- Installation du système et programme 'Hello World'
- Contrôler les moteurs du robot
- Construction du robot
- Reconnaissance des couleurs
- Le robot suit une ligne droite
- Le robot suit un circuit
- Le robot trouve la ligne

Architecture, conception et gestion de projet

Tout au long du développement, nous avons principalement utilisé deux technologies parmi d'autre, pour le développement de notre produit, ces technologies sont:

- **Lejos** : un firmware pour les briques programmables Lego Mindstorms.
- **Java** : un langage de programmation basé sur des classes, orienté objet.
- **Eclipse** : environnement de développement intégré.

Le temps de développement de notre projet était censé être de sept mois, mais avec la pandémie de coronavirus nous avons fait notre développement en cinq mois seulement

Architecture, conception et gestion de projet

Pour tester nos programmes et notre robot, on a utilisé des grands circuits colorés imprimés sur papier.

Pendant le développement, on a rencontré un problème un peu difficile, et que nous avons passé beaucoup de temps, c'était la capacité du robot à trouver la ligne à suivre après être sorti du circuit ou s'il était mis à l'extérieur du circuit.

Pour résoudre ce problème, nous avons fait en sorte que le robot recherche la ligne en effectuant des tours de virage incrémentiels jusqu'à ce qu'il trouve la ligne, et lorsqu'il le fait, il se repositionne pour continuer à suivre la ligne.

Programmation

```
public void lookForLine() throws RemoteException, MalformedURLException, NotBoundException {
    sampleRGBValue = new float[3];
    Boolean turnDebut = true;
    long timeToTurn = 500, time = System.currentTimeMillis(), tStart = time, turnTime = 600, turnTimeStart = time;
    int i=0,motorSpeed = 200;
    String detectedColorName;
    // While the brick is detecting
    while(Button.ESCAPE.isUp()) {
        // Detect a color and get the RGB values
        this.sampleRGBValue = sampleProvider.fetchSample();
        this.sampleRGBValue[0] = this.sampleRGBValue[0] * 256f;
        this.sampleRGBValue[1] = this.sampleRGBValue[1] * 256f;
        this.sampleRGBValue[2] = this.sampleRGBValue[2] * 256f;

        // Get the name of the detected RGB value
        detectedColorName = Color.getColor(this.sampleRGBValue, toAvoidColor, toFollowColor);
    }
}
```

```
// Print the detected color
LCD.drawString(detectedColorName, 0, 1);
if(detectedColorName.equals(toFollowColor.getName())) {
    largeRegulatedMotorDroit.stop(true);
    largeRegulatedMotorGauche.stop(true);
    break;}
else {
    time = System.currentTimeMillis();
    if(time-tStart > timeToTurn) {
        if(turnDebut) {
            turnTimeStart = time;
            turnDebut = false;}
        if(time-turnTimeStart < turnTime) {
            largeRegulatedMotorGauche.setSpeed(0);
            largeRegulatedMotorDroit.setSpeed(motorSpeed);}
        else {
            tStart=time;
            i++;
            if(i == 4) {
                timeToTurn *= 2;
                i = 0;}
            turnDebut = true;}}
    else {
```



```
        largeRegulatedMotorGauche.setSpeed(motorSpeed);
        largeRegulatedMotorDroit.setSpeed(motorSpeed);}

    largeRegulatedMotorGauche.forward();
    largeRegulatedMotorDroit.forward();}}

while(Button.ESCAPE.isUp()) {
    largeRegulatedMotorGauche.setSpeed(motorSpeed);
    largeRegulatedMotorGauche.forward();
    largeRegulatedMotorDroit.setSpeed(motorSpeed);
    largeRegulatedMotorDroit.backward();
    this.sampleRGBValue = sampleProvider.fetchSample();
    this.sampleRGBValue[0] = this.sampleRGBValue[0]*256f;
    this.sampleRGBValue[1] = this.sampleRGBValue[1]*256f;
    this.sampleRGBValue[2] = this.sampleRGBValue[2]*256f;
    detectedColorName = Color.getColor(this.sampleRGBValue, toAvoidColor, toFollowColor);
    if(detectedColorName.equals(toAvoidColor.getName())) {
        Delay.msDelay(300);
        largeRegulatedMotorGauche.stop(true);
        largeRegulatedMotorDroit.stop(true);
        return;}}}
```

COVID-19

Quel impact les mesures contre le COVID-19 ont eu sur le projet ?

Dès le début de confinement, on n'a pas pu travailler sur le projet, ça était à cause de difficulté de travailler à la maison, la charge des autre module, et des projets, le manque de matériel pour développer et tester le robot (le robot était chez un binôme et le circuit et les battery pour tester étaient chez l'autre)

Dans des conditions normales, nous aurons travaillé plus sur le robot, rendu notre algorithm meilleur, aurait testé notre robot plus et le robot aurait suivi la ligne mieux et en douceur.

Conclusion

Dans ce projet, nous avons fait nos premiers pas dans le monde de la robotique, nous avons appris comment développer un petit robot ce qui est un suiveur de ligne et nous avons également développé nos compétences en Java, en utilisant une autre librairie de ce langage (Lejos) .

Une deuxième version de notre projet serait, un suiveur de ligne plus rapide qui fonctionne en ligne droite et un code plus propre.

Si nous devions refaire le projet, nous ne changerons rien.