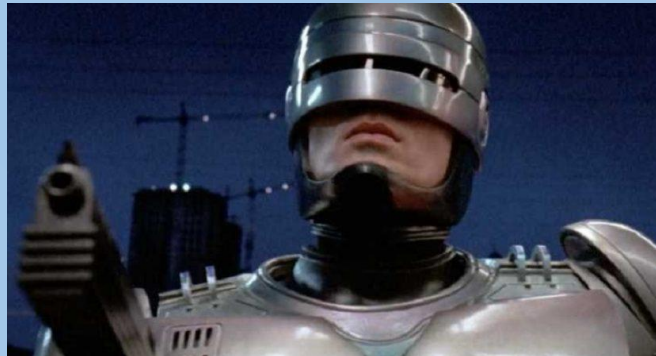


Projet Long 2018-2019

ROBOCOP

*Aouessar Oussama
Charif Mohammed-Achraf*



Université Paris Diderot Paris 7



- Introduction :

À quel problème répond votre projet ?

Notre projet est un robot suiveur de ligne, sa fonction principal est de prendre une ligne de couleur à définir et se contente de la suivre. quelle que soit les courbes, quelle que soit le chemin, il y a un algorithme derrière qui corrige le robot et essaye de le remettre au droit chemin.

Pourquoi est-ce que ce problème a un intérêt ?

Aujourd'hui le monde a besoin de plus en plus de robot qui ne fatigue pas et qui ne se repose pas. Alors notre "Robocop" est là pour accomplir des tâches de transport ou comme un drone de reconnaissance qui va parcourir un chemin et dessiner une map. il peut être utilisé à des fins variées et peut répondre vos besoins.



- *Présentation succinctes des fonctionnalités :*

Quelles sont ses fonctionnalités principales ?

Robocop a pour fonctionnalité de prendre une ligne de couleur choisi par l'utilisateur et de suivre cette ligne quelle que soit la difficultés et les courbes. Il retrouvera toujours son chemin au cas ou il sort de la ligne.

Quels sont les scénarios d'utilisation principaux ?

Il y a plusieurs scénarios, il peut être utilisé comme transporteur d'un point de départ A a un point d'arrivé B, par exemple "Amazon" utilise ce genre de robot pour préparer les commandes des clients. Il peut aussi être utiliser pour parcourir un chemin à des fin de reconnaissance, comme il peut être utilisé pour faire des courses.



- architecture, conception et gestion de projet

Le projet est divisé en 3 grandes parties, LineColors, RGB et LineFollower

- Line Colors :

Cette classe a pour but d'apprendre, de stocker et de tester les couleurs choisies par l'utilisateur dans le menu du robot. Elle est directement liée à la classe RGB et permet de calculer le barycentre d'une classe de couleurs.

On a aussi une fonction qui permet de tester et reconnaître les couleurs qui ont été apprises. On l'utilise pour vérifier si le capteur marche bien et si les couleurs ont été stockées correctement.

- RGB :

Cette classe définit une couleur par trois valeurs: Red, Green et Blue. puis compare les entiers renvoyant la distance.

```
public double compareInteger(int r, int g, int b) {  
    int dist_r = r - red;  
    int dist_g = g - green;  
    int dist_b = b - blue;  
  
    if(Math.abs(dist_r) >= 55 || Math.abs(dist_g) >= 55 || Math.abs(dist_b) >= 55) {  
        return Double.MAX_VALUE;  
    }  
  
    return Math.sqrt(Math.sqrt(dist_r) + Math.sqrt(dist_g) + Math.sqrt(dist_b)); //La distance entre les  
}
```

- Line Follower :

C'est la classe principal du projet ou ce trouve tous l'algorithme de suivie de ligne, la gestion des moteurs et la partie graphique qui s'occupe de la communication avec l'utilisateur et le choix des couleurs.

```
public int colorToFollow() {  
    // Construction de ma liste de couleurs  
    String[] colorSelect = new String[cl.getColorID()];  
    for (int i = 0; i < colorSelect.length; i++)  
        colorSelect[i] = ""+i;  
  
    // Menu  
    TextMenu modeMenu = new TextMenu(colorSelect, 1, "Color to follow");  
    while(Button.ESCAPE.isUp()) {  
        LCD.clear();  
        int mode = modeMenu.select();  
        switch(mode) {  
            case 0:  
                LCD.clear();  
                return 0;  
            case 1:  
                LCD.clear();  
                return 1;  
  
            default:  
                break;  
        }  
        LCD.clear();  
        break;  
    }  
    return -1;  
}
```

Une partie de l'algorithme du line follower:

```
while(Button.ESCAPE.isUp()){  
    if(cl.getColorSeenID() == colorToFollow){  
        motorSetSpeed(speed_droite, speed_gauche);  
        Motor.A.forward();  
        Motor.C.forward();  
    }else{//pas la meme couleur donc il va chercher la couleur  
        if(turn){//tourner a gauche -- turn=true c'est gauche --  
            int c = 0;  
            int cc = 0;  
            while(true){  
                motorSetSpeed(450, 450);  
                Motor.A.backward();  
                Motor.C.forward();  
                cc++;  
                if(cl.getColorSeenID() == colorToFollow){  
                    turn = false;  
                    c = 1;  
                    break;  
                }  
                if(cc == 100) break;//ancienne valeur 100  
            }  
            if(c == 1){  
                continue;  
            }  
            c = 0;  
            cc = 0;  
            while(true){  
                motorSetSpeed(450, 450);  
                Motor.A.forward();  
                Motor.C.backward();  
                cc++;  
                if(cl.getColorSeenID() == colorToFollow){  
                    turn = true;  
                    c = 1;  
                    break;  
                }  
                if(cc == 130) break;//ancienne valeur 150  
            }  
            if (c == 1){  
                continue;  
            }  
        }  
    }
```

- Programmation :

```
while(Button.ESCAPE.isUp()){
    if(cl.getColorSeenID() == colorToFollow){
        motorSetSpeed(speed_droite, speed_gauche);
        Motor.A.forward();
        Motor.C.forward();
    }else{//pas la meme couleur donc il va chercher la couleur
        if(turn){//tourner a gauche -- turn=true c'est gauche --
            int c = 0;
            int cc = 0;
            while(true){
                motorSetSpeed(450, 450);
                Motor.A.backward();
                Motor.C.forward();
                cc++;
                if(cl.getColorSeenID() == colorToFollow){
                    turn = false;
                    c = 1;
                    break;
                }
                if(cc == 100) break;//ancienne valeur 100
            }
            if(c == 1){
                continue;
            }
            c = 0;
            cc = 0;
            while(true){
                motorSetSpeed(450, 450);
                Motor.A.forward();
                Motor.C.backward();
                cc++;
                if(cl.getColorSeenID() == colorToFollow){
                    turn = true;
                    c = 1;
                    break;
                }
                if(cc == 130) break;//ancienne valeur 150
            }
            if (c == 1){
                continue;
            }
        }
    }
}
```

```
if (c != 1){
    continue;
}
}else{
    int c = 0;
    int cc = 0;
    while(true){
        motorSetSpeed(450, 450);
        Motor.C.backward();
        Motor.A.forward();
        cc++;
        if(cl.getColorSeenID() == colorToFollow){
            turn = true;
            c = 1;
            break;
        }
        if(cc == 100) break;
    }
    if(c == 1){
        continue;
    }
    c = 0;
    cc = 0;
    while(true){
        motorSetSpeed(450, 450);
        Motor.C.forward();
        Motor.A.backward();
        cc++;
        if(cl.getColorSeenID() == colorToFollow){
            turn = false;
            c = 1;
            break;
        }
        if(cc == 130) break;
    }
    if (c == 1){
        continue;
    }
}
}
```



- Déroutement du projet / git :

Combien de commits avez-vous fait ? Avec quelle fréquence ? Pourquoi ? Quels sont les scénarios d'utilisation principaux ?

Pour la gestion du git, on a divisé notre dépôt en 2 répertoires différents (SandBox et Production) on développe dans le répertoire "SandBox" et dès qu'on fini une itération on l'ajoute au répertoire "Production" qui contient notre projet principale et la version final à rendre. On a pas fait beaucoup de commit parce que tous nos commit pushed contient des gros morceaux de code et des fonctionnalités complète et fini. En ce qui concerne la fréquence d'utilisation de git, tout dépend de la fonctionnalité en cours de développement. Par exemple le commit qui contient l'algorithme du LineFollower a pris plus de temps que la gestion de couleurs (environ 1 mois d'écart entre les deux commit).



- Conclusion :

D'abord merci à tous nos enseignants, c'était une expérience de programmation très intéressante, et différente, ça change un peu du quotidien de l'université où on reste collé devant nos écrans. Certes on a rencontrés beaucoup de difficultés au début surtout l'installation de l'environnement de développement Lejos sur nos machines parce qu'on a des systèmes d'exploitations différents. Mais ce qui nous a le plus posé de problèmes c'est la vitesse. Notre robot maintenant même avec une vitesse un peu juste il arrive à suivre une ligne et permet de surmonter n'importe quel obstacle de courbes et s'adapter à tous les terrains. On l'a testé un peu partout dans le campus de l'université et il arrive à suivre une ligne.

Encore merci, pour votre attention, et pour cette expérience à ne pas oublier.

Aouessar - Charif