

SOUTENANCE PROJET LONG MINDSTORM NXT



Souleymane DIALLO
Duc Anh NGUYEN

Plan

I. Introduction

II. Fonctionnalités

III. Architecture, conception et gestion de projet

IV. Programmation

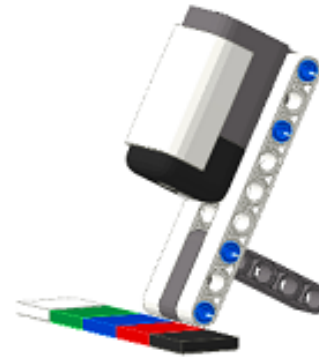
V. Conclusion

Introduction

- Automatisation de certains procédés manuels
 - dans les industries
 - dans la vie de tous jours: les lignes de transports (bus, métro, etc..)
- Rapidité
- Sécurité
- Productivité
- Performance

Fonctionnalités

- Lire des couleurs
- Apprendre des couleurs
- Reconnaître une couleur : méthode Gaussienne
- Suivi d'une ligne à partir de sa couleur



Sénarios d'utilisation



Architecture

- Module de gestion des couleurs:
 - ColorRGB
 - GaussianParam
 - Sample
 - ReadWriteColors
 - ColorSelector
- Module de suivi de ligne
 - LineFollower

Conception

- Prise en main: familiarisation avec l'API LEJOS
 - Installation
 - Configuration
 - Tests
- Echantillonnage
 - Capture de couleurs
 - Méthodes d'échantillonnage
 - Sauvegarde des couleurs
- Suiveur de ligne
 - Ligne droite
 - Ligne courbée

Gestion de projet

- 1 séance de 2h toutes les semaines ou 2
- Dépôt Git
- Communication régulière
- Séance de projet long de chaque 2 ou 3 semaines

Programmation

```
77
48 public void move2(int leftPow, int rightPow){
49     MotorPort.C.controlMotor(leftPow, 1);
50     MotorPort.A.controlMotor(rightPow, 1);
51 }
52
53
54 public void followLine(){
55     //timer = new Timer(1000, this);
56     speed_gauche = BASESPEED;
57     speed_droite = BASESPEED;
58     int angle = 0;
59     boolean turn = true;
60     on = false;
61     while(!Button.ESCAPE.isDown()){
62         /* When robot saw the main color, he goes left*/
63         Color color = colorSelector.getColorFromSensor();
64         ColorRGB c = new ColorRGB(color.getRed(),color.getGreen(),color.getBlue());
65         if(colorSelector.isColorFollowed(c)){ /* When robot is in the line*/
66             if (on) {
67                 on = false;
68             }
69             angle=0;
70
71             if(speed_gauche != speed_droite){
72                 speed_gauche = speed_droite= Math.min(speed_gauche, speed_droite);
73             }
74             //move();
75             if(speed_gauche < MAXSPEED && speed_droite < MAXSPEED){
76                 speed_gauche += SPEED_STEP;
77                 speed_droite += SPEED_STEP;
78             }
79             move2(speed_gauche,speed_droite);
80         } else { /* When robot is outline*/
81             if (!on) {
82                 speed_gauche = BASESPEED;
83                 speed_droite = BASESPEED;
84                 on = true;
85             }
86
87             turn(angle,turn);
88             turn = !turn;
89             angle +=5;
90         }
91     }
92 }
93
94 ...
```

```
244
245 public void followLine7(){
246     int err = 0;
247     int i =1;
248     while(!Button.ESCAPE.isDown()){
249         /* When robot saw the main color, he goes left*/
250         Color color = colorSelector.getColorFromSensor();
251         ColorRGB c = new ColorRGB(color.getRed(),color.getGreen(),color.getBlue());
252         if(colorSelector.isColorFollowed(c)){ /* When robot is in the line*/
253             move2(80,10);
254         } else { /* When robot is outline*/
255             move2(10,80);
256         }
257     }
258 }
259
260 public void turn(int angle, boolean left){
261     //pilot.setRotateSpeed(50);
262     //pilot.setTravelSpeed(150);
263     if(left){
264         move2(0,70+angle);
265     }else{
266         move2(70+angle,0);
267     }
268 }
```

Conclusion

- Une expérience enrichissante
- De nouveaux acquis:
 - Gestion de projet
 - Programmation sur robot
- Une version plus performante
 - Rapidité
- Difficulté
 - USB 3.0 → Bluetooth
 - échantillonnage

Avez-vous des questions?