

YUGANSAN Yogaratnam
ZERZERI Nabil



MINDSTORMS®

PROJET MINDSTORM

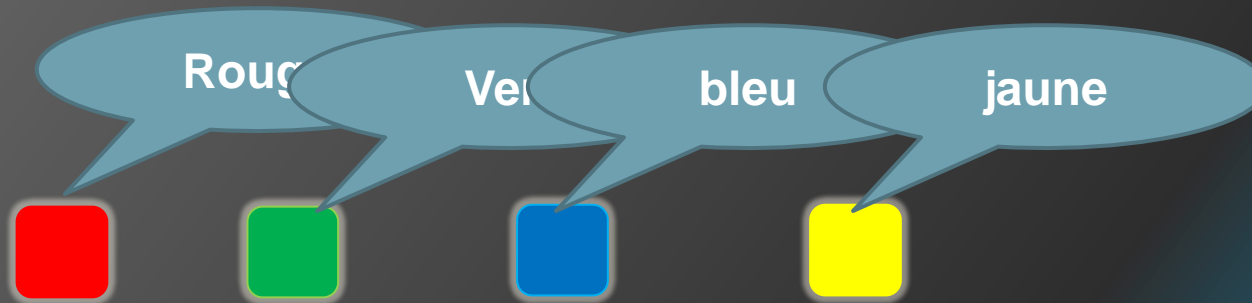
Sommaire

- I. Introduction
- II. Fonctionnalités
- III. Architecture, conception et gestion de projet
- IV. Programmation
- V. Conclusion

Introduction

Suivre une ligne

Reconnaissance de couleurs

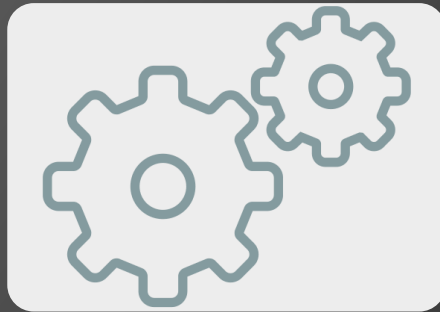


Introduction

Intérêt



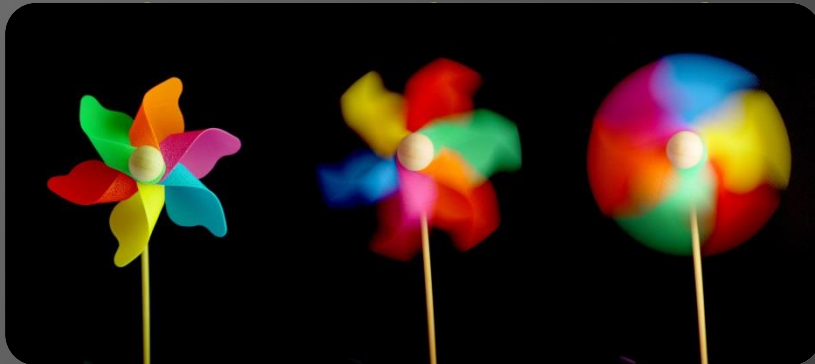
reconnaissance



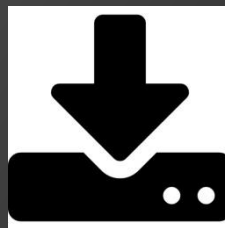
automatisation

Fonctionnalités

Détection des couleurs



Gestion des mouvements



Gestion des données
(enregistrer les couleurs)

Fonctionnalités

● Scenario :

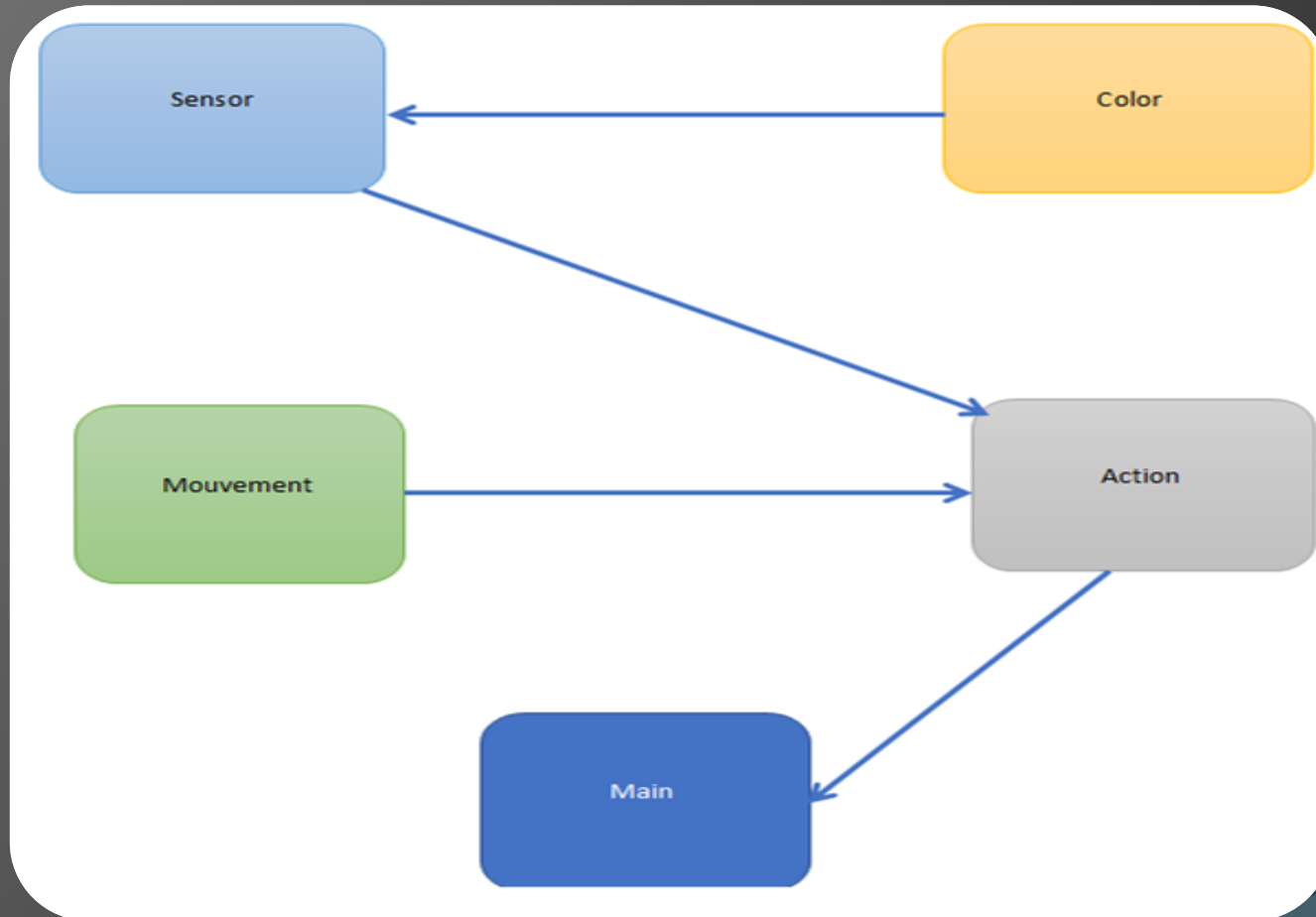
Utilisation comme jeu pour enfant



Composante d'un robot automatisé



Architecture



Conception

- ❑ **Aller tout droit**
- ❑ **Reconnaitre les couleurs**
- ❑ **Suivre une ligne droite**
- ❑ **Suivre un circuit simple**
- ❑ **Suivre un circuit complexe :**
 - intersection,
 - Demi-tour / fin de circuit
 - Choisir un chemin parmi plusieurs

Gestion de projet



- Technique:
 - Construction lego
 - Java lejos
 - Utilisation de l'interface NXT



- Durée :



- 1an

- Répartition :

- **module par module en synchronisation**



Gestion de projet

- ❑ Test de l'algorithme effectué sur les circuits de l'université
- ❑ Difficultés Technologiques :
 - précision du capteur, corriger par plusieurs scan et un intervalle de couleur plus proche des scans effectué
- ❑ Difficultés Technique:
 - Conception de l'algorithme le plus efficace
- ❑ Difficultés avec le Capteur lightSensor trop proche du sol, corriger par un suport plus gros et qui se tient droit

Programmation

```
eclipse-workspace - ProjLong/src/action/MoveLeftRightAction.java - Eclipse
File Edit Source Refactor Navigate Search Project LeJOS NXJ Run Window Help

ColorMemory.java MoveLeftRightAction.java

23     }
24 }
25
26
27 private boolean whileLeftSide(){
28     boolean isColorCrossed=false;
29     boolean colorScaned=false;
30
31     while(!isColorCrossed){
32         movement.moveForward();
33         movement.moveRotationLeft();
34         if(lightSensor.getPosAndVerifie(positionCouleur[0])){
35             colorScaned=true;
36             movement.setSpeed();
37         }else if(lightSensor.getPosAndVerifie(positionCouleur[1])){
38             movement.turnAroundLeft();
39             System.out.println("Apres le demi tour");
40         }
41         else if(colorScaned){
42             isColorCrossed=true;
43             movement.decreaseSpeed(movement.getLeftMotor());
44         }else{
45             movement.increaseSpeed(movement.getRightMotor());
46             movement.decreaseSpeed(movement.getLeftMotor());
47             nbTimeDontSee++; //ajoute
48             if(nbTimeDontSee>NBDEDEGRE){
49                 searchColor();
50             }
51             movement.moveRotationLeft();
52         }
53     }
54     return true;
55 }
56
```

Conclusion

- ❑ V2:
 - Gestion de circuit plus complexe (plusieurs couleurs, obstacle sur le chemin, mouvement plus fluide et rapide)
 - Apprentissage des circuits
- ❑ Robot mieux construit
- ❑ Appris à utiliser des bibliothèques, à gérer des imprécisions des tests, utilisation d'un robot automatisé

