

Projet Long

Mindstorms

2017-2018

Université Paris Diderot

1/13

Thomas Guignard
Slimane Siroukane

Sommaire

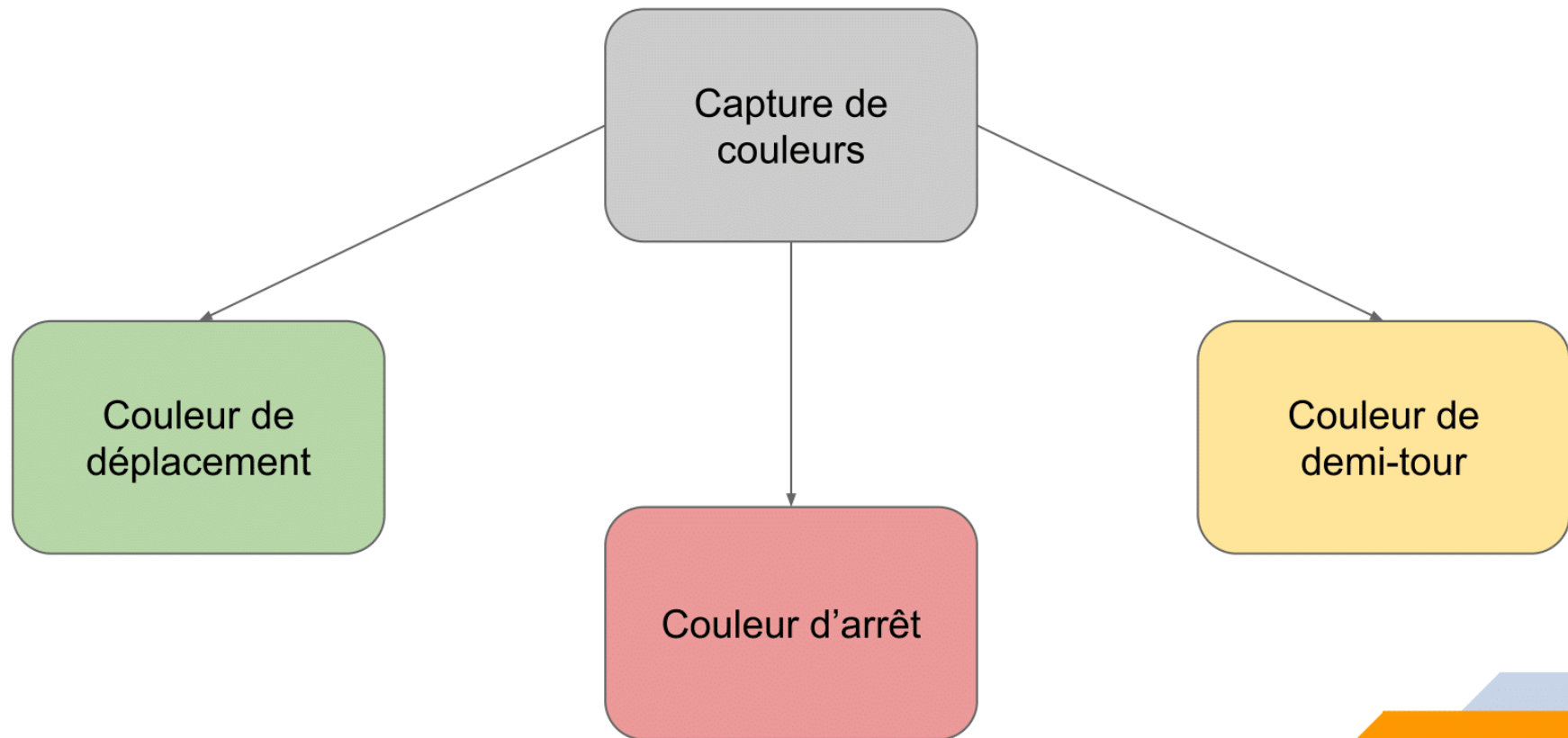
1. Introduction
2. Fonctionnalités
 - A. Capture de couleurs
 - B. Parcours d'un circuit
3. Architecture & Gestion de projet
4. Programmation
5. Conclusion

Introduction



Fonctionnalités

A. Capture de couleurs

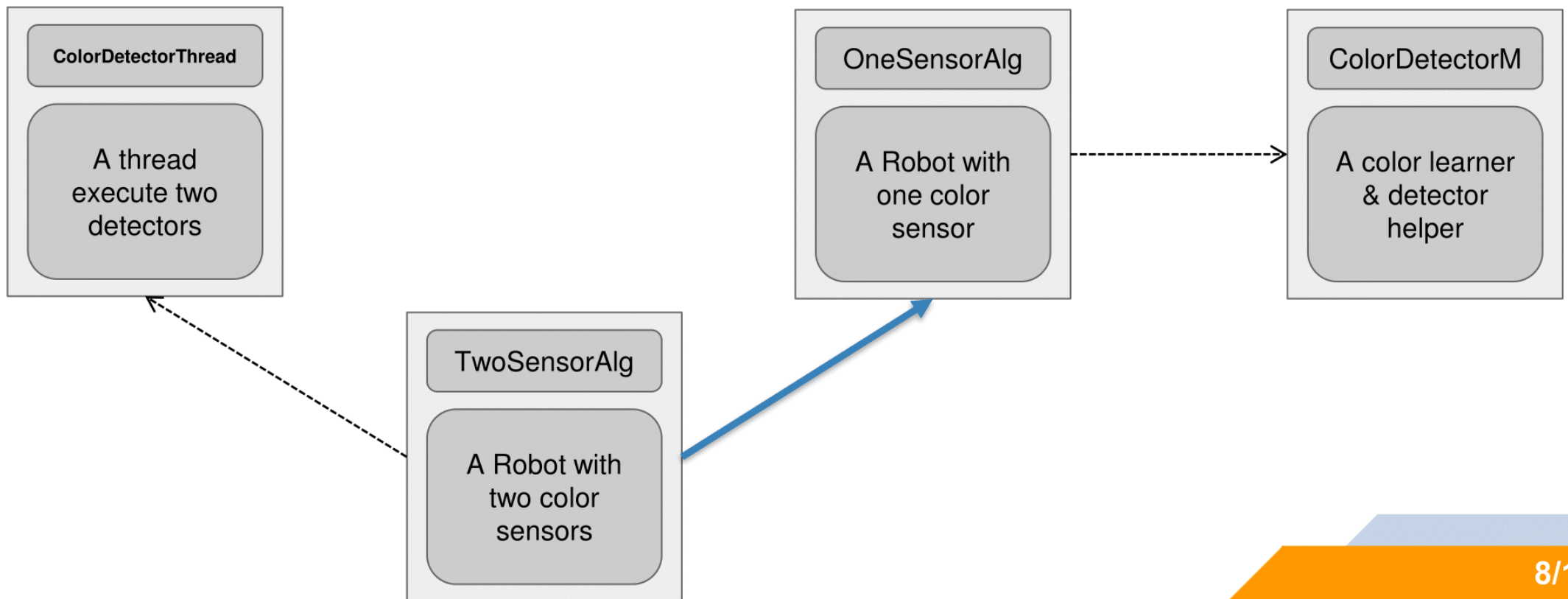


B. Parcours d'un circuit

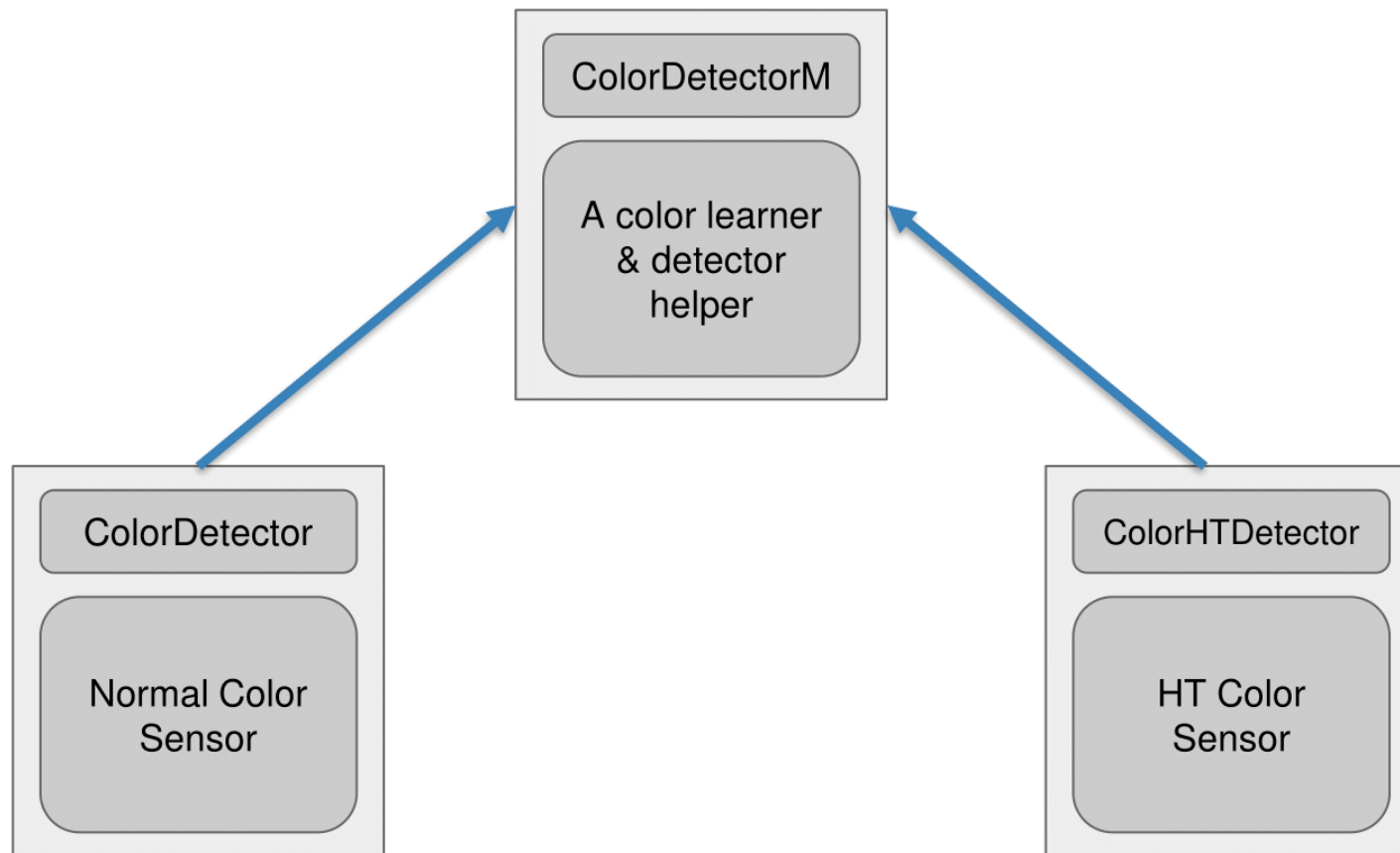
- Le robot **tourne** lorsqu'il ne détecte plus une des couleurs capturées
- Lors d'un **carrefour** : le robot se déplace tout droit

Architecture & Gestion de Projet

Architecture



Architecture



Gestion de projet

- Equipe : 2 développeurs
- Durée : 7 mois
- Organisation :
 - travail personnel avec le robot 1 semaine sur 2
 - rencontre de l'équipe chaque semaine, échange du robot
 - présentation aux enseignants toutes les 2 semaines

Difficultés rencontrées

- Matériel : version antérieure du LEGO Mindstorms (NXT)
- Capture de couleurs : différenciation des couleurs

Programmation

```
public static void main(String[] args) {
    colorDetectors = new ColorDetectorM[2];
    colorDetectors[0] = new ColorHTDetector(
        nbr_colors, new ColorHTSensor(SensorPort.S3));
    colorDetectors[1] = new ColorDetector(
        nbr_colors, new ColorSensor(SensorPort.S2));

    colorDetectorThreads = new ColorDetectorThread[2];
    colorDetectorThreads[0] = new ColorDetectorThread(
        results, 0, colorDetectors[0]);
    colorDetectorThreads[1] = new ColorDetectorThread(
        results, 1, colorDetectors[1]);

    for (int i = 0; i < nbr_colors; i++) {
        for (int j = 0; j < colorDetectors.length; j++) {
            ColorDetectorM.stockColorScreen(
                i, j, colorDetectors[j]);
        }
    }

    colorDetectorThreads[0].start();
    colorDetectorThreads[1].start();
    walkForward();
}
```

```
do {
    if (getCurrentAction() == GO_FORWARD) {
        walkForward();
    } else if (getCurrentAction() == ROTATE_LEFT) {
        rotate(error);
    } else if (getCurrentAction() == ROTATE_RIGHT) {
        rotate(-error);
    } else if (getCurrentAction() == GO_BACK) {
        if (goBackCommand == BACK_CAMMAND) goBack();
        else break;
    } else {
        rotate(-error);
    }
} while (!Button.ESCAPE.isDown());

stopWalking();
colorDetectorThreads[0].interrupt();
colorDetectorThreads[1].interrupt();
}
```

Conclusion