

ROBOT Mindstorms®

MASTER1 PARIS DIDEROT INFORMATIQUE PROJET LONG

Yasmine HAMDANE

Amadou DOUMBIA

Brice DUGUAY

Plan

1. Introduction.
2. Présentation des fonctionnalités.
3. Architecture, conception et gestion du projets .
4. Programmation.
5. Conclusion.

Introduction

1. Présentation du robot
Mindstorms®

Son intérêt

2. But du Projet

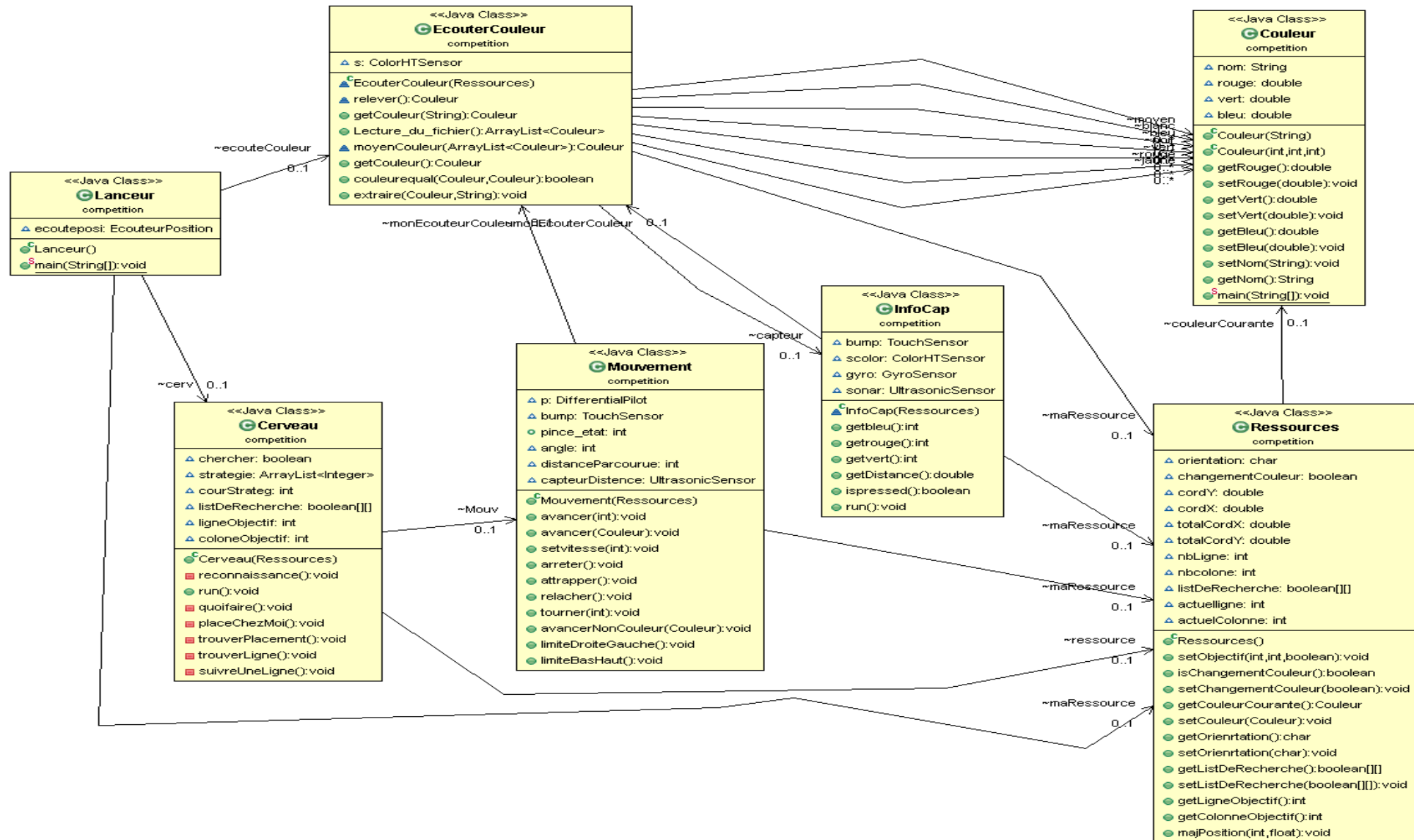
3. Prise En main Lejos



Les grandes fonctionnalités du robot

1. Détecter une ligne/colonne
2. Suivre une ligne/colonne
3. Attraper un palet
4. Poser à la maison
5. Relâcher le palet
6. Retourner sur le tapis pour récupérer d'autres palets
7. Resource- Orientation.
8. Aller à une couleur données

Architecture/Conception



Conception des composants

1. Un module responsable du lancement du robot
2. Cerveau s'occupe du choix de la stratégies à suivre
3. Un module responsable d'écouter sans cesse la couleur sur laquelle il est, et tester s'il est bloqué.
4. Module qui gère tous les mouvement du robots
5. Module mettant à jour tous les capteur, et les positions au fur et à mesure que le robot se déplace

Distribution du travail

Une liste de toutes les méthodes était établie, chacun s'est attribué le même nombre de méthodes à coder.

Programmation

1. Le langage de programmation utilisé pour notre robot est leJOS, Java pour la plateforme NXT. La
2. Documentation disponible abondante sur le site internet "Lejos"
3. Pour l'écriture du code, nous avons privilégié l'environnement de développement Eclipse


```

private void suivreUneLigne() {
    Mouv.setvitesse(10);
    int deltaRotation = 5;
    boolean gauche = true;
    Couleur couleurinitiale = Mouv.monEcouteurCouleur.getCouleur();
    while (!Mouv.bump.isPressed()
        || Mouv.capteurDistance.getDistance() > 15) {
        if (!Mouv.monEcouteurCouleur.getCouleur().nom
            .equals(couleurinitiale.getNom())) {
            if (gauche == true) {
                rotationGauche(deltaRotation);

                if (!Mouv.monEcouteurCouleur.getCouleur().nom
                    .equals(couleurinitiale.getNom())) {

                    rotationDroite(deltaRotation);
                }
            }
        }
    }
}

```

```

else {

    rotationDroite(deltaRotation);
    if (!Mouv.monEcouteurCouleur.getCouleur().nom
        .equals(couleurinitiale.getNom())) {
        rotationGauche(deltaRotation);
    }

    }
    deltaRotation += 2;
    gauche = !gauche;
}

```

Conclusion

1. Ce qu'on a appris
2. Les améliorations qu'on pourrait apporter