



# Plan

Introduction.

Présentation des fonctionnalités.

Architecture, conception et gestion du projets .

Programmation.

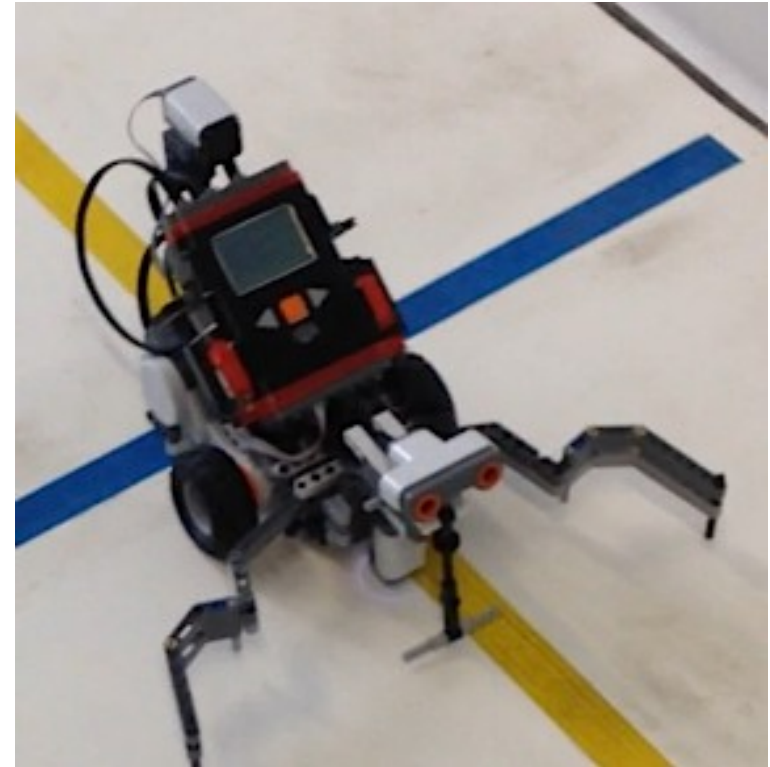
Conclusion.

# Introduction

1. Présentation du robot Mindstorms®

*Son intérêt*

2. But du Projet

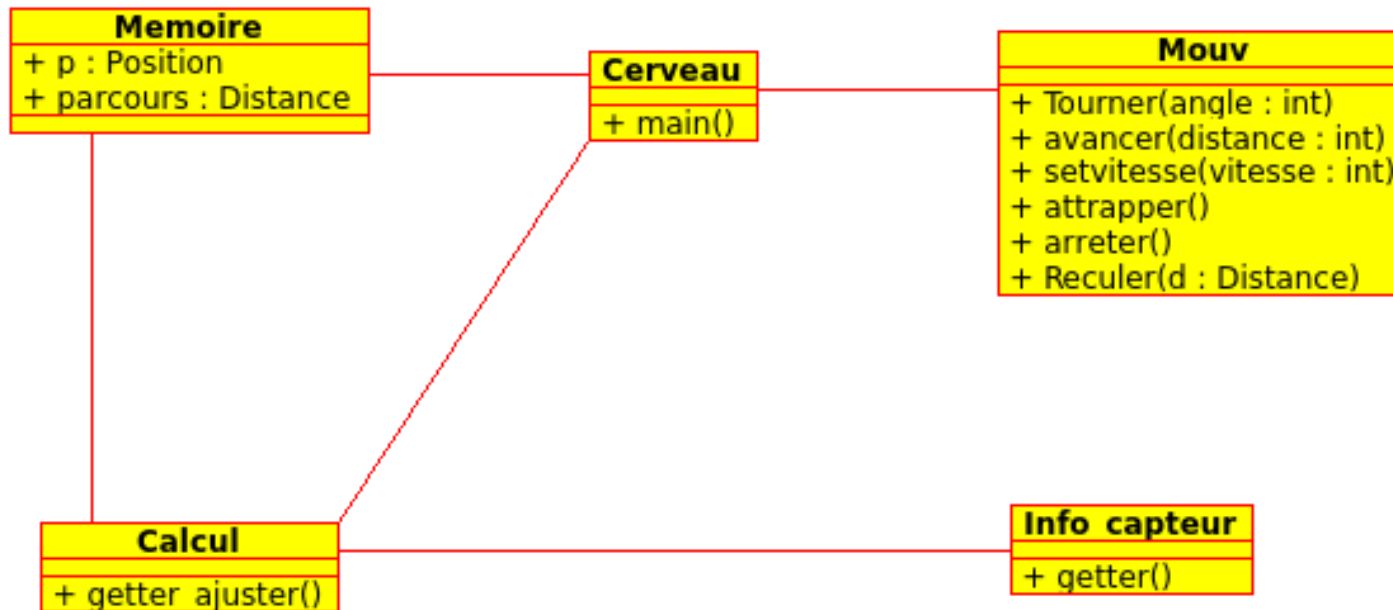


# Les grandes fonctionnalités du robot

1. Détecter une ligne/colonne
2. Suivre une ligne/colonne
3. Attraper un palet
4. Le déplacer vers un emplacement délimité par une ligne noire
5. Relacher le palet
6. Retourner sur le tapis pour récupérer d'autres palets

# Architecture/Conception

diagramme de classe:



# Conception des composants

1. Un module reponsable du lancement du robot et du choix de la strategie à suivre
2. Un module responsable d'écouter sans cesse la couleur sur lequel il est, et tester s'il est bloqué.
3. Module qui gère tous les mouvement du robots
4. 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.monEcouleurCouleur.getCouleur();
    while (!Mouv.bump.isPressed()
        || Mouv.capteurDistance.getDistance() > 15) {
        if (!Mouv.monEcouleurCouleur.getCouleur().nom
            .equals(couleurinitiale.getNom())) {
            if (gauche == true) {
                rotationGauche(deltaRotation);

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

                    rotationDroite(deltaRotation);
                }
            }
        }
    }
}

```

```

else {

    rotationDroite(deltaRotation);
    if (!
Mouv.monEcouleurCouleur.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