

# Nao Collecteur de déchets

# Introduction

- **Objectif Initial du projet:**

Notre projet consiste à réaliser un programme qui permet au robot NAO de rechercher, collecter et mettre les déchets dans une poubelle.

Que pouvons nous faire pour préserver notre environnement?

- **Solutions que rapporte notre projet:**

Une solution, la collecte automatique de déchets recyclable dont le but est de:

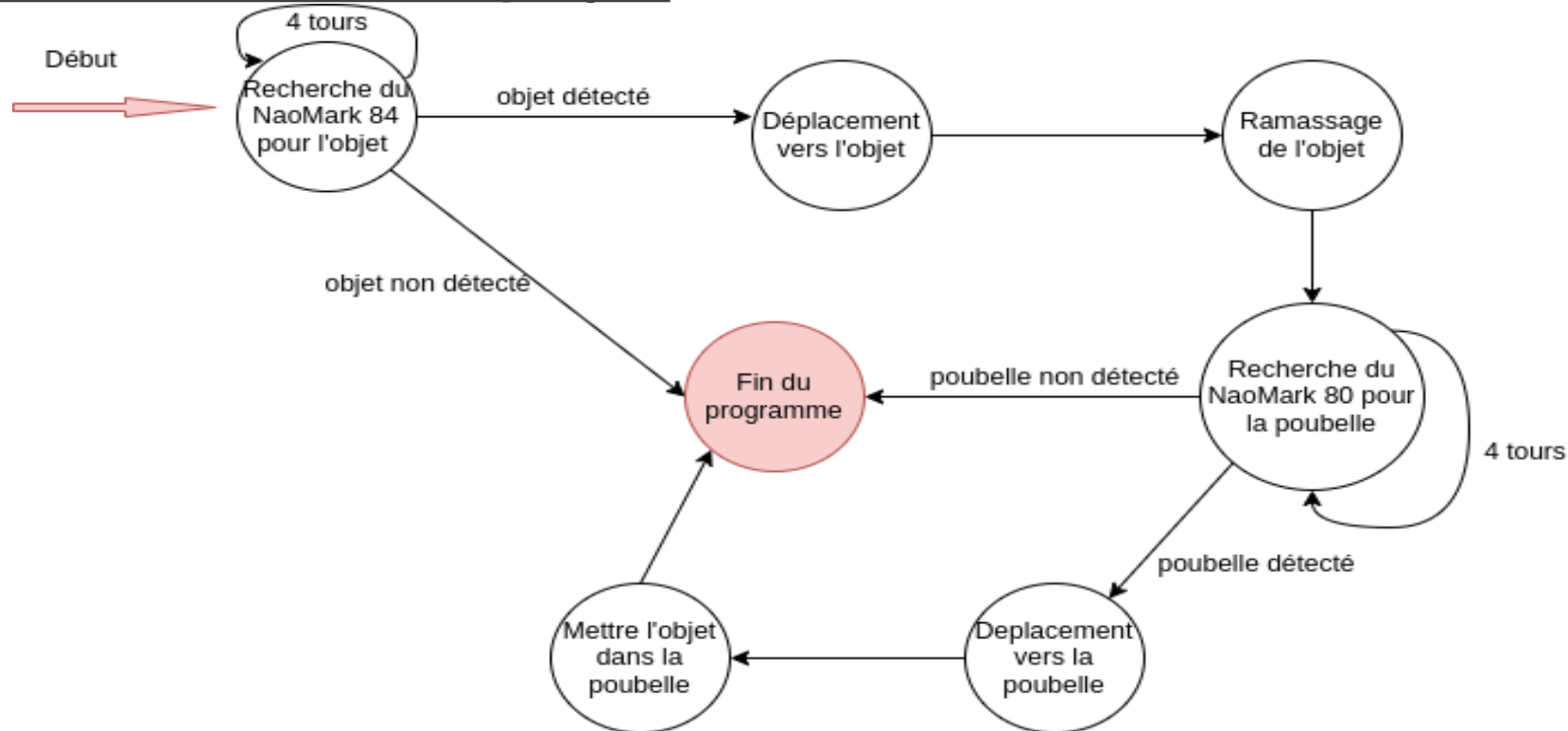
1. Lutter contre la pollution.
2. réduire les couts liés à la collecte de déchets.

# Introduction

► Demo:

# Architecture conception et gestion de projet:

## Fonctionnement du projet:



# Architecture conception et gestion de projet:

## ► Etapes de réalisation du Projet:

**Le projet a été décomposé en deux parties:**

- Déplacement
- Recherche

# Architecture conception et gestion de projet:

## ► compétences techniques:

Les compétences techniques nécessaires pour la réalisation du projet sont:

- Le langage Python.
- L' API Naoqi pour le robot Nao.
- Familiarisation avec le robot.

## 📋 Repartition du travail:

Dans notre cas le travail n'a pas été réparti entre les membres du groupe, on a travaillé ensemble durant tout le processus de développement et de tests.

# Architecture conception et gestion de projet:

## Processus de test:

Les tests ont été effectués sur le robot à chaque étape de développement.

## Difficultés rencontrées:

- Installation du logiciel de simulation webot.
- Limitation de l'API NAOQi qui fonctionne seulement avec la version 2.7 de Python
- Détection d'obstacle. **solution:** sensors du Nao qui lui permettent de ne pas rentrer en collision avec les obstacles.
- Entraînement d'un réseau de neurone pour la reconnaissance des objets. pas résolu car le processus demandait beaucoup de temps, mais à la place nous avons utilisé les Naomark.

# Programmation

```
Fonction recherche_NaoMark(NaoMark_Number)
    compteur := 0
    findBottle := Faux
    baisser_tete() // pour regarder en bas.
    markData = memory_service.getData(memValue) // recuperer les données du NaoMark en mémoire.

    Tant que( taille(markData) = 0 et compteur < 4)
        Si (findBottle = Faux)
            MovementRotation() // fonction pour faire tourner le robot de 90° sur lui-même en bougeant les mains.
        Sinon
            rotation() // fonction pour faire tourner le robot de 90° sur lui-même sans bouger les mains.
        FINSI

        markData = memory_service.getData(memValue)
        compteur := compteur + 1

    SI (taille(markData) >=2 )
        POUR i de 1 jusqu'à taille(markData)
            SI markData[i] = 84 // NaoMark 84 pour identifier la bouteille
                CoordonneeNaoMark = recupere_Coordonnee() // pour recuperer les coordonnées de la position du NaoMark
                theta = arcTang(CoordonneeNaoMark.y / CoordonneeNaoMark.x)
                moveTo(1.4*CoordonneeNaoMark.x-0.10 , CoordonneeNaoMark.y-0.10, -theta-0.10)
            SINON SI markData[i] = 80 // NaoMark 84 pour identifier la poubelle
                CoordonneeNaoMark = recupere_Coordonnee() // pour recuperer les coordonnées de la position du NaoMark
                theta = arcTang(CoordonneeNaoMark.y / CoordonneeNaoMark.x)
                moveTo(1.4*CoordonneeNaoMark.x-0.10 , CoordonneeNaoMark.y-0.10, -theta-0.10)
        SINON
            ECRIRE(Objet non trouvé)
```



# Conclusion:

## La réalisation de ce projet nous a permis d'apprendre:

- le langage PYTHON.
- L'utilisation de l'API NAOQI (pour le NAO).
- L'utilisation du simulateur choregraphe qui nous a accompagné durant toute la réalisation du projet.

La version 2 de notre projet sera de programmer un robot qui reconnaîtra les objets par leurs formes et leurs couleurs.

Si on nous demandait de refaire le projet on aurait passé plus de temps sur la reconnaissance d'objets que sur l'apprentissage de déplacement.