

# Plan de développement

## 1. Conception générale du système

- **Objectif** : Concevoir le fonctionnement global de l'automate en définissant les différents états et les conditions de transition entre eux.
- **Étapes** :
  - **État 0 : Prendre et marquer le premier palet.** Le robot va à une position prédéfinie, ramasse le palet et le dépose dans le camp adverse.
  - **État 1 : Rechercher.** Le robot cherche un palet sur la table (utilisation de capteurs pour la détection).
  - **État 2 : Se déplacer.** Une fois le palet trouvé, le robot se déplace vers lui.
  - **État 3 : Prendre le palet.** Le robot ramasse le palet avec ses pinces.
  - **État 4 : Marquer.** Le robot déplace le palet vers le camp adverse et le dépose.
  - **Conditions de transition** :
    - Transition de **Prendre et marquer le premier palet** à **Rechercher** après avoir déposé le premier palet.
    - Transition de **Rechercher** à **Se déplacer** lorsque le capteur détecte un palet.
    - Transition de **Se déplacer** à **Prendre le palet** lorsque le robot atteint la position du palet.
    - Transition de **Prendre le palet** à **Marquer** après avoir ramassé le palet.
    - Transition de **Marquer** à **Rechercher** après avoir déposé le palet.

## 2. Répartition des tâches par membre du groupe :

### Flora : Gestion des états et des transitions

- **Tâche** :
  - Implémenter la gestion des états avec un automate à états finis.
  - Utiliser un 'switch' et une boucle 'while' pour gérer les différents états du robot.
  - S'assurer que les transitions entre les états sont correctes en fonction des conditions (capteurs, actions terminées).
  - **Livrable** :
    - Fonction principale (boucle FSM) qui contrôle le passage entre les états selon les événements déclenchés par les capteurs et les actions du robot.
- **Étapes** :
  - Créer une structure pour la machine à états avec les états **RECHERCHE**, **SE\_DEPLACER**, **PRENDRE\_PALET**, **MARQUER**.
  - Chaque état doit appeler une fonction spécifique (développée par les autres membres du groupe).

### Victor : Développement de l'état 'Prendre et marquer le premier palet'

- **Tâche :**
  - Implémenter la logique pour l'état initial où le robot se déplace vers une position prédéfinie, ramasse le premier palet, et le dépose dans le camp adverse.
  - Gérer les mouvements avec le 'MovePilot' et les pinces pour ramasser et déposer le palet.
  - **Livrable :**
    - Fonction qui guide le robot vers la position prédéfinie pour le premier palet et qui gère le ramassage et la dépose.
- **Étapes :**
  - Utiliser les coordonnées prédéfinies pour diriger le robot vers le palet.
  - Ramasser le palet avec les pinces, se déplacer vers le camp adverse et déposer le palet.

### Alexander : Développement des états 'Rechercher' et 'Se déplacer'

- **Tâche :**
  - Implémenter la fonction associée à l'état **Rechercher** où le robot utilise son capteur ultrason pour détecter la présence d'un palet.
  - Implémenter la fonction associée à l'état **Se déplacer** pour que le robot se déplace vers le palet détecté.
  - **Livrable :**
    - Fonction qui gère la recherche d'un palet.
    - Fonction qui déplace le robot vers le palet une fois détecté.
- **Étapes :**
  - Utiliser le capteur pour détecter les palets.
  - Utiliser le capteur afin de pouvoir potentiellement esquiver le robot adverse.
  - Utiliser 'MovePilot' pour diriger le robot vers la position du palet.

### Thomas : Développement des états 'Prendre le palet' et 'Marquer'

- **Tâche :**
  - Implémenter la fonction associée à l'état **Prendre le palet**, où le robot utilise ses pinces pour ramasser le palet.
  - Implémenter la fonction associée à l'état **Marquer**, où le robot déplace le palet vers le camp adverse et le dépose.
  - **Livrable :**
    - Fonction pour ramasser le palet avec les pinces.
    - Fonction pour diriger le robot vers le camp adverse et déposer le palet.
- **Étapes :**
  - Ramasser le palet en fermant les pinces.
  - Déplacer le robot vers la zone de marquage (camp adverse) et ouvrir les pinces pour déposer le palet.

### 3. Phase d'intégration

- **Objectif** : Intégrer les différentes fonctions développées dans l'automate à états finis.
- **Tâches** :
  - S'assurer que les transitions entre les états sont bien définies et que le robot passe d'un état à un autre en fonction des événements.
  - Tester les fonctions d'état individuellement, puis tester l'ensemble des états en chaîne pour assurer la fluidité du comportement du robot.
  - **Livrable** :
    - Une version fonctionnelle du programme où le robot passe d'un état à un autre de manière fluide et sans erreurs.

### 4. Phase de test et validation

- **Objectif** : Tester le robot dans un environnement réel et vérifier son comportement dans chaque état.
- **Tâches** :
  - Tester chaque état séparément pour s'assurer que chaque action est exécutée correctement (déplacement, recherche, ramassage, marquage).
  - Tester les transitions entre les états, et vérifier que les capteurs fonctionnent correctement pour déclencher les transitions.

### 5. Optimisation et amélioration

- **Objectif** : Optimiser le comportement du robot pour qu'il soit plus rapide et plus précis.
- **Tâches** :
  - Optimiser les mouvements du robot pour minimiser les temps de trajet.
  - Ajuster les transitions pour réduire les temps d'attente.
  - **Livrable** :
    - Version finale optimisée du programme.
    - Documentation finale du projet.