Idées miniprojet :

* Faire bouger le robot avec le son : Micro (siffler génère une chorégraphie)
* Labyrinthe : Caméra (ajouter couleurs pour marche/arrêt)
* Détection de couleur : Caméra (Par exemple : rouge -> avance, bleu -> recule)
* Contrôle du robot par pression : Si on touche le robot derrière il accélère : IMU ?
* Robot qui fait de l’escalade : Sur plan incliné, observe les prises et choisit la plus proche puis se déplace : Caméra et IMU (Avancé : éviter les obstacles)
* Niveau (LED s’allume selon la direction de l’inclinaison) : IMU

Idée choisie :

Route : détection de ligne noire et la suivre. Lorsque la caméra détecte une lumière rouge il s’arrête jusqu’à ce qu’il détecte une lumière verte et avance (lumière orange + contrôle par capteur). Lumière -> LED ou couleur sur papier ou image sur téléphone. Le robot avance et ne recule pas. (Il accélère quand il descend sur un plan incliné). Si le robot s’arrête, les leds rouges s’allument (Clignotants quand il tourne)

**Etape 1 :** Définir

Commencez par définir votre projet sur papier :

• quels capteurs vous allez utiliser : est-ce que vous êtes au clair sur les caractéristiques ? Est-ce que vous devez valider des aspects ?

Capteur de distance infrarouge et caméra (IMU)

• quelle application vous voulez faire : quels choix sont délicats, où est-ce qu’il peut y avoir des problèmes ?

Quelle démo finale ? Faisable par vidéo interposée

• quelles ressources vous voulez utiliser : quelles représentations de nombres, mémoire, FPU, DSP, périphériques ?

**Etape2:**

Organiser Avant d’écrire le code et les algorithmes en détail, prenez le temps de planifier SUR PAPIER l’architecture de votre code, avec la construction de différentes librairies, gestion correcte et optimale des Threads, etc.

Sur la base du TP4 :

* ProcessImage : Garder celui du TP4 et ajouter la détection de couleur, modifier la détection et le suivi de la ligne
* ProcessDistance : Thread de traitement pour les 8 capteurs de distance infrarouge

(

* Accélération moteur

)

Priorité : Capteur de distance -> Détection de couleurs -> Détection de ligne -> Moteurs