

IronCrousty (Christopher, Elias Antoun, Louis, Maxence, Laïssa, Aya)

IronCrousty

Objectif

Dans le cadre de l'UE Projet Robotique, nous devons fabriquer un robot capable de récupérer un objet et de le déposer ailleurs dans une arène. Afin de réaliser cela, nous devons nous répartir dans plusieurs pôles (informatique, électronique et mécanique) pour qu'aucun de aspect de la fabrication ne soit oublié. La communication entre les personnes des différents pôles est donc primordiale pour la bonne conduite du projet.

Répartition des rôles :



- Responsable informatique : Maxence
- Pôle informatique : Maxence, Elias, Christopher
- Responsable électronique: Elias
- Pôle électronique : Christopher, Aya, Laïssa, Elias
- Responsable mécanique : Louis
- Pôle mécanique : Louis, Aya, Laïssa
- Chef de projet : Elias

Séance 1 (12/02/26)

- Répartition des rôles et création du diagramme de Gantt (Christopher)

https://docs.google.com/spreadsheets/d/1wC_1XbupPaG7TZ1aQCpUpkzV-nBctYzU/edit?gid=83363326#gid=83363326

Pôle informatique

Test et vérification du bon fonctionnement des composants du kit à l'aide de programmes Arduino (Maxence, Elias, Aya)

- Test des deux moteurs KTECH MS4015-V3
- Test du moteur Dynamixel
- Test de la pince 3551
- Test des deux capteurs à ultrasons HC-SR04

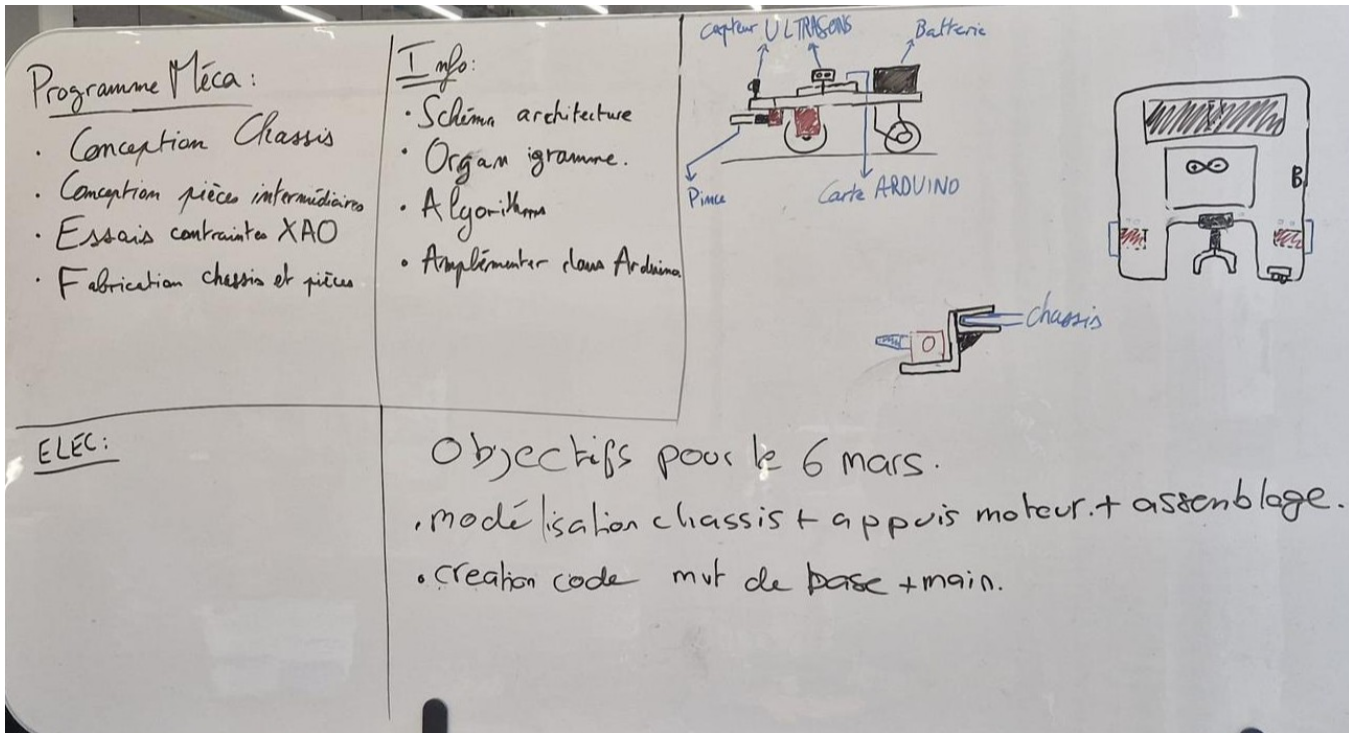
Création du Git : https://github.com/Maxence-Santos/Projet_ROB3_2_S6

Pôle mécanique

- Réflexion et conception générale de l'architecture du robot. Réalisation d'un croquis au tableau permettant la définition de l'emplacement des différents composants : batterie, carte Arduino, capteurs à ultrasons et pince. Proposition de plusieurs solutions de conception, puis mise en accord sur une solution commune. (Lousi, Laïssa, Aya, Elias, Christopher, Maxence)

Fixation des objectifs mécaniques pour la suite du projet (Louis, Laïssa, Aya)

- Modélisation du châssis
- Conception des appuis moteurs et des pièces intermédiaires
- Étude des contraintes mécaniques
- Fabrication du châssis et des pièces



Pôle électronique

Réalisation du câblage des composants du kit et vérification des connexions (Aya, Maxence, Elias)

- Câblage des deux moteurs KTECH MS4015-V3 via le bus CAN
- Câblage du moteur Dynamixel
- Câblage de la pince 3551
- Câblage des deux capteurs à ultrasons HC-SR04
- Branchement de l'alimentation et vérification générale du circuit

Séance 2 (19/03/26)

- Remplissage et amélioration du Wiki (Aya)

Pôle informatique

- Finalisation des tests du kit et vérification du bon fonctionnement de l'ensemble des composants. (Maxence, Elias)
- Début de la réflexion et mise en place de l'organigramme général du programme de contrôle du robot. (Christopher)

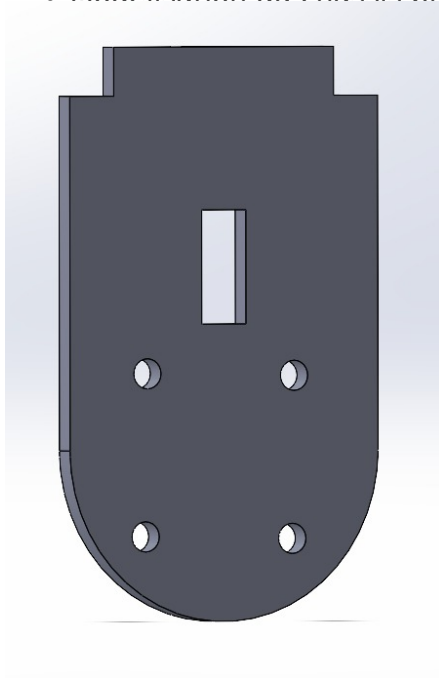
Pôle électronique

- Soudage du câble d'alimentation du moteur Dynamixel en vue de son intégration dans le circuit électrique du robot. (Elias)

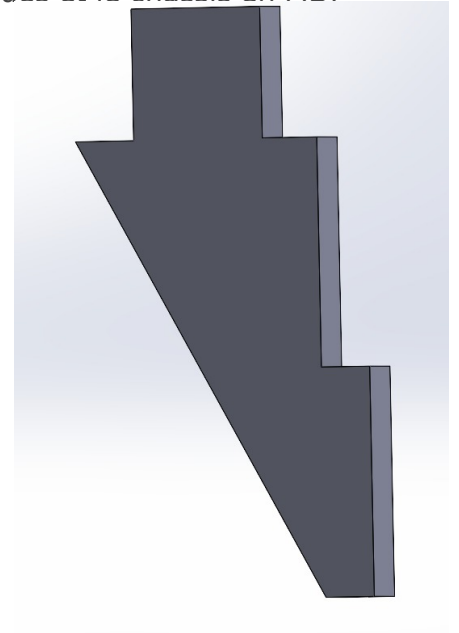
Pôle mécanique

Début de la modélisation des différentes pièces nécessaires à l'assemblage du robot, en fonction des matériaux et procédés de fabrication retenus :

- Modélisation du châssis principal en bois MDF 6mm, destiné à être découpé au laser (Aya) pour la découpe laser (Laïssa)

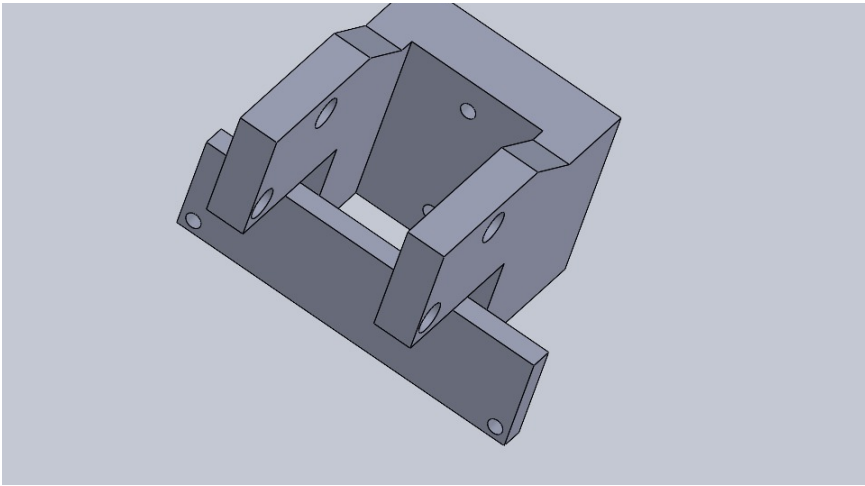


Raccord moteur roue et châssis
support fixe au raccord

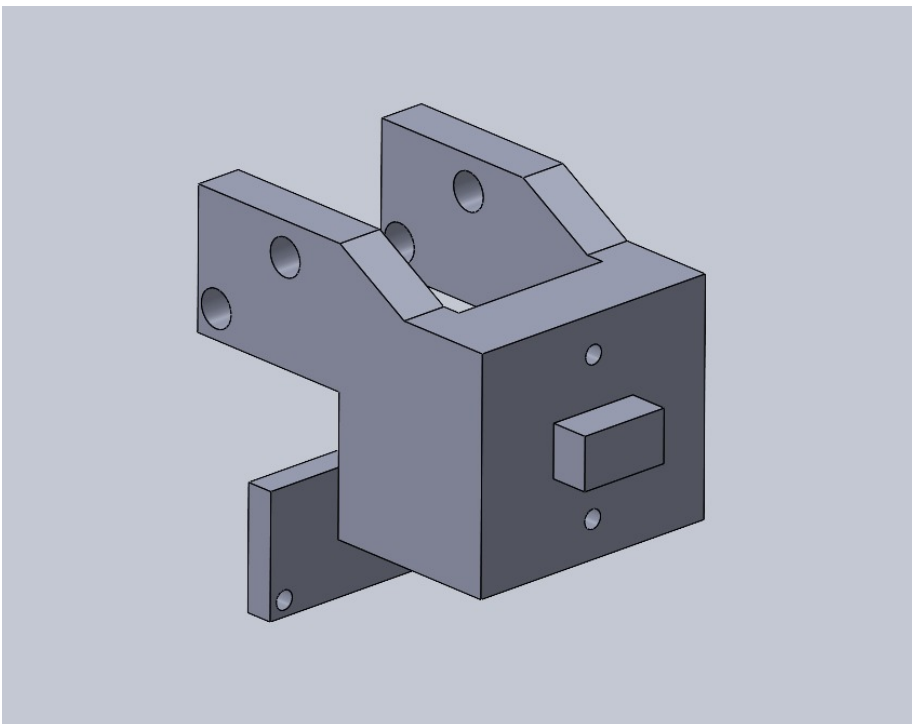


Triangle de

- Modélisation 3D d'une pièce intermédiaire pour l'intégration de la pince, conçue pour être fabriquée par impression 3D (Louis)



Pièce intermédiaire pince-chassis

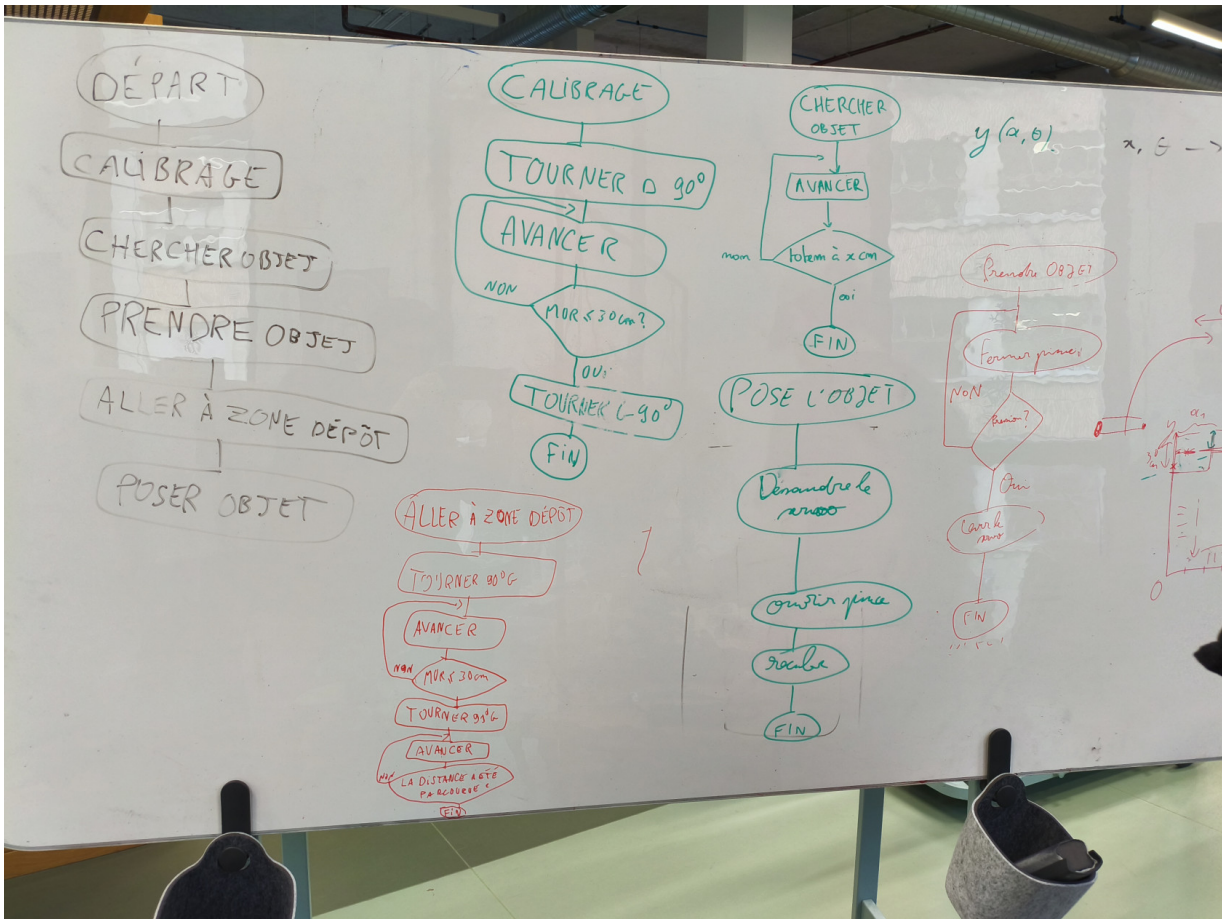


Séance 3 (02/04/26)

- Remplissage et amélioration du Wiki (Aya)

Pôle informatique

- Réalisation de l'organigramme principal (Christopher, Maxence, Elias) ainsi que des sous-programmes.

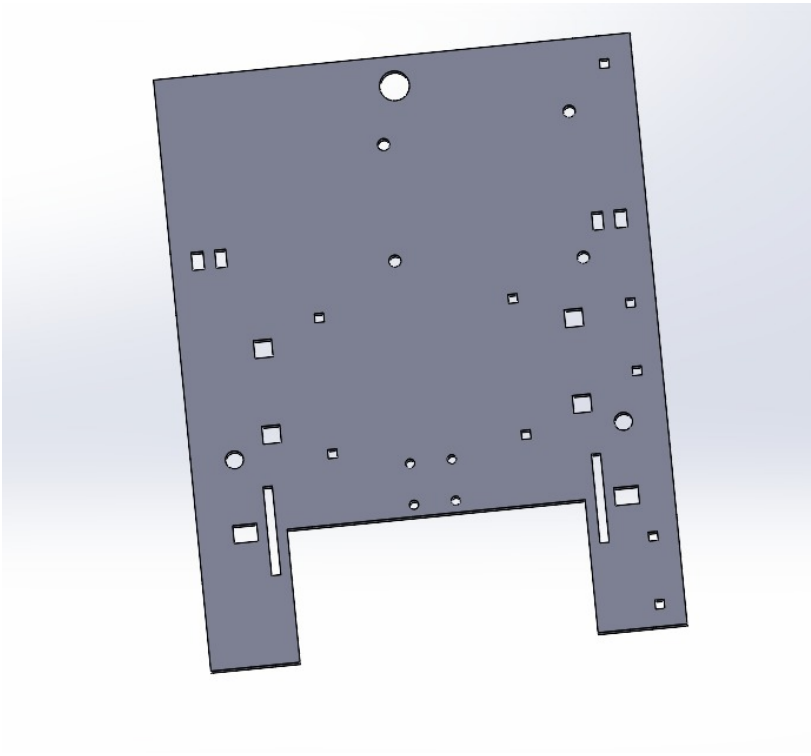


- Codage des fonctions pour tourner et avancer : elle prend en argument la distance à parcourir, l'angle à avoir et la vitesse à laquelle les roues doivent tourner. Elle fait d'abord tourner le robot de l'angle demandé, puis le fait avancer de la distance demandée. (Christopher)
- Codage de la fonction prendre objet (Elias)

Pôle mécanique

Modification et affinement du châssis principal (Laïssa)

- Ajout de trous et de détails techniques
- Forme générale inchangée

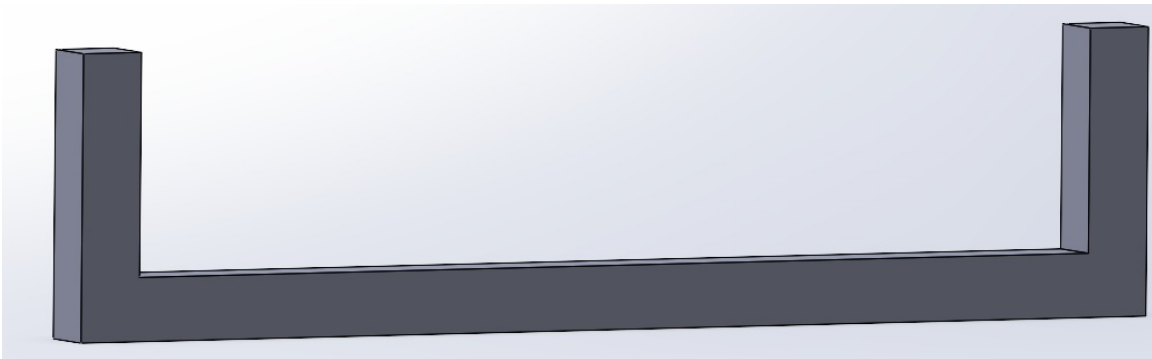


Modélisation des pièces de maintien de la batterie (Laïssa)

- Fixation stable sur le châssis
- Conception prenant en compte le poids de la batterie

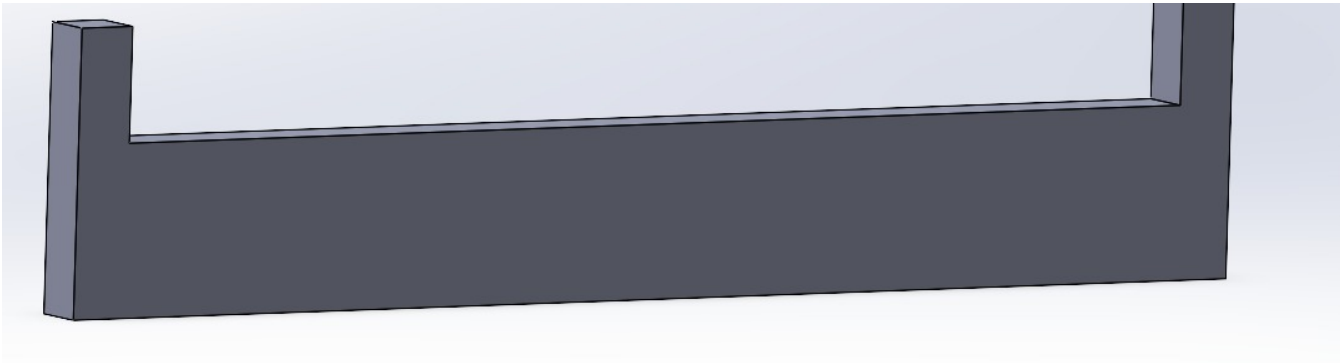
Modélisation d'un support pour capteur à ultrasons (Elias)

- Destiné à la détection de mur latéral



Support

poids de batterie



Maintien de batterie

Séance 4 (03/04/26)

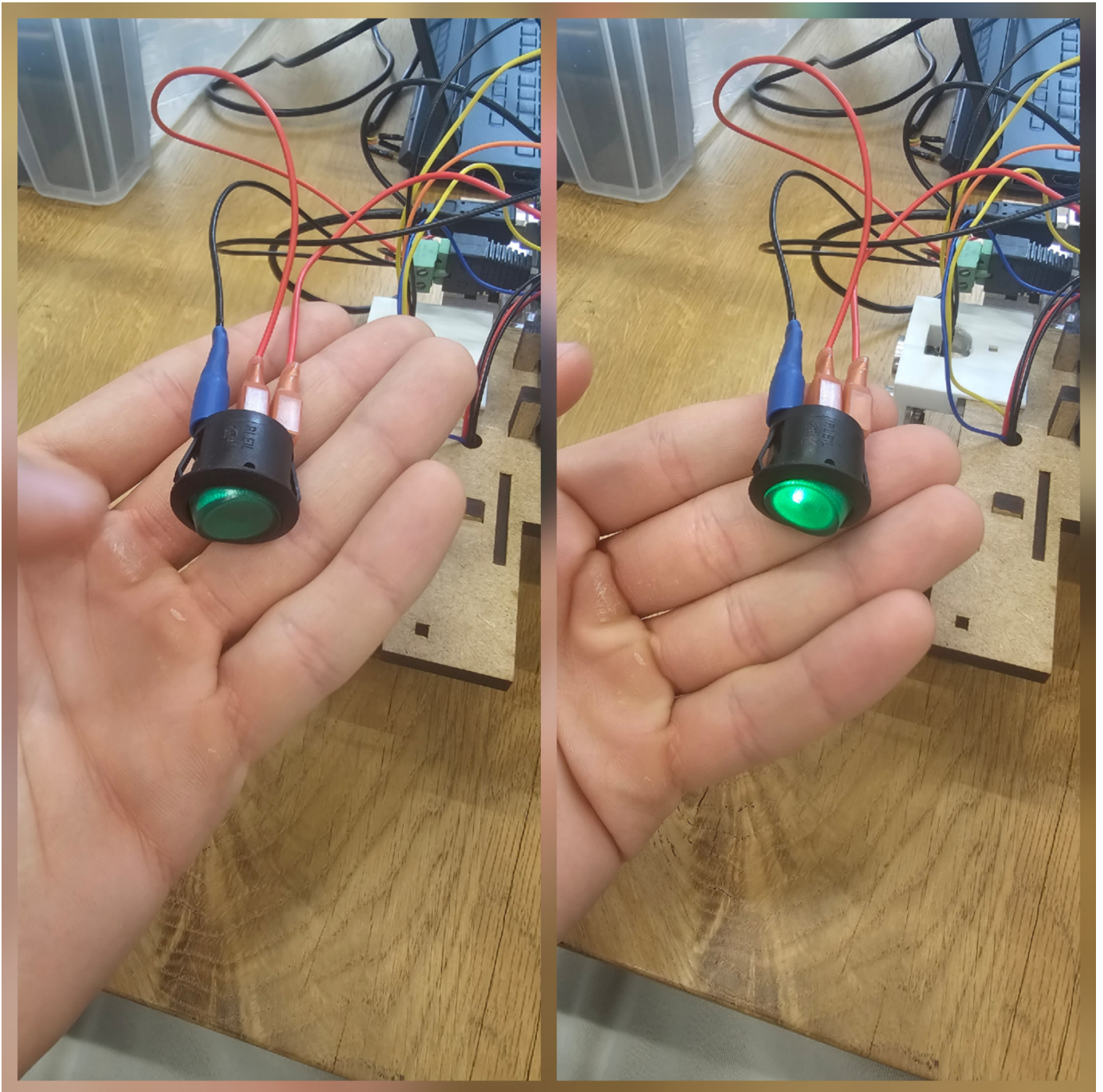
- Remplissage et amélioration du Wiki (Aya)

Pôle mécanique

- Intégration de l'ensemble des pièces modélisées dans un assemblage global (Laïssa)
- Vérification et amélioration de la précision des trous et des détails sur les différentes pièces (Aya)
- Ajustements effectués afin d'assurer la compatibilité et le bon assemblage entre les composants (Louis)

Pôle électronique

- Soudage d'un bouton qui s'allume (Elias)



Séance 5 (10/04/26)

- Remplissage et amélioration du Wiki (Aya)

Pôle mécanique

- Découpe de toutes les pièces du robot MDF selon les plans modélisés sur SolidWorks (Laïssa, Louis)
- Vérification de la conformité des pièces obtenues avec les dimensions prévues (Laïssa, Louis)

Séance 6 (16/04/26)

- Remplissage et amélioration du Wiki (Aya)
- Le diagramme de Gantt a été complété maintenant que les étapes intermédiaires sont plus claires (Christopher)

Pôle informatique

- Codage de la fonction `calibrage()`. Cette fonction est faite pour que le robot se place à 30cm du mur DC car on sait que le totem l'est aussi. Pour ce faire, on le fait pivoter vers la droite, avancer jusqu'à être à distance souhaitée, puis pivoter vers la gauche. (Christopher)
- Codage de la fonction `chercher_objet()` et de la librairie `init_moteurs` afin de ne pas avoir à initialiser les moteurs et la connexion CAN dans chaque code qui utilise les moteurs. (Maxence)

Pôle mécanique

- Assemblage et ajustement entre la pince et la pièce intermédiaire puis entre la pièce intermédiaire et le moteur Dynamixel (Louis)
- Montage du robot sans la pince (Laïssa)

Pôle électronique

- Soudage de câbles plus robustes (Elias)

Séance 7 (20/05/26)

Pôle informatique

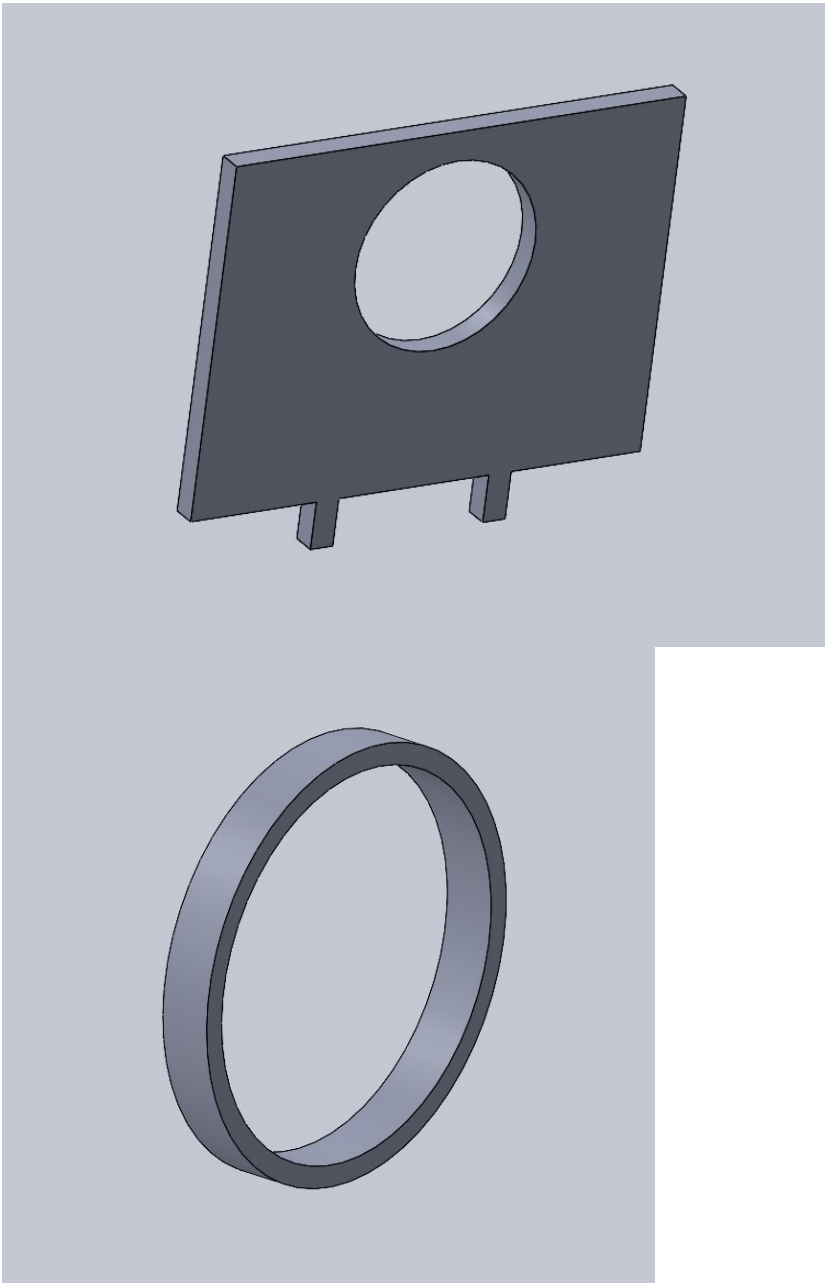
- Amélioration de la fonction `chercher_objet` (Maxence)

Séance 8 (21/05/26)

- Remplissage et amélioration du Wiki (Aya)

Pôle mécanique

- Montage complet du robot incluant la pince (Laïssa, Louis)
- Modélisation et Montage de la pièce support pour le bouton démarrage. Rajout d'une bague qui va réduire le jeu entre le bouton et la pièce (Louis)



Pôle informatique

- Test et amélioration de la fonction `deplacer()` (Christopher)
- Amélioration de la fonction `deplacer()`, elle se base maintenant sur deux fonctions : `tourner` et `avancer`. (Christopher)

Séance 9 (21/05/26)

Pôle informatique

- Création de la fonction `corriger_droite()` qui corrige la trajectoire du robot en le faisant rester à une certaine distance du mur de droite à plus ou moins 10%. (Christopher)
- Création des fonctions pour détecter un objet à une certaine distance (devant ou à droite). (Christopher)
- Achèvement de la fonction `calibrage_initial()` (anciennement `calibrage()`). (Christopher)

Séance 10 (26/05/26)

Pôle informatique

- Reconsidération totale du programme de déplacement du robot (Elias)

Séance 11 (27/05/26)

Pôle informatique

- Implémentation, dans `chercher_objet.cpp`, d'une logique de déplacement et de détection d'objet à l'aide des capteurs ultrasoniques, tout en ajoutant un système de correction de trajectoire ("correcteur") afin de maintenir un déplacement stable malgré les écarts mécaniques des moteurs (Maxence, Elias)
- Intégration de l'odométrie pour estimer la distance parcourue par le robot à partir de la rotation des roues, ce qui permet d'évaluer précisément la position de l'objet dans l'arène et d'améliorer la navigation (Maxence, Elias)
- Programmation du contrôle de la pince servo dans `pince.cpp` afin de saisir et relâcher les objets automatiquement (Maxence, Elias)
- Développement de la logique de retour vers la zone de départ en exploitant les données d'odométrie et les corrections de trajectoire pour garantir un déplacement plus précis et fiable dans `retour.cpp` (Maxence, Elias)

Pôle électronique

- Remaniement du câble management (Elias)

Pôle mécanique

- Ajout d'une tige essentielle entre les supports moteurs-chassis, ce qui a réduit les écarts de coaxialité des moteurs et a considérablement amélioré les mouvements du robot (Laïssa)

Séance 12 (28/05/26) Évaluation

- Remplissage et amélioration du Wiki (Louis, Laïssa)
- Création de la notice du robot 'IronCrousty' (Aya)

Pôle informatique

- Amélioration et finalisation des programmes de déplacement et de capture du totem

Notice d'utilisation du robot

1. Présentation du robot

IronCrousty est un robot autonome développé dans le cadre de l'UE Projet Robotique. Son objectif est de localiser un objet (totem) dans une arène, de le saisir, puis de le déposer dans une zone cible désignée, sans intervention humaine.

Équipe	Christopher, Elias , Louis, Maxence, Laïssa, Aya
Chef de projet	Elias
Responsable info	Maxence
Responsable élec	Elias
Responsable méca	Louis

2. Composants matériels

Le robot est constitué des éléments suivants :

- 2 moteurs de roues KTECH MS4015-V3 (pilotés via bus CAN)
- 1 moteur Dynamixel (contrôle de la pince)
- 1 pince modèle 3551
- 2 capteurs à ultrasons HC-SR04 (détection avant et latérale)
- 1 carte Arduino (carte de contrôle principale)
- 1 batterie avec système de maintien imprimé en 3D
- 1 châssis principal en bois MDF 6 mm découpé au laser
- Pièces intermédiaires en MDF 3 mm et impression 3D
- 1 bouton lumineux vert (démarrage du programme)

3. Mise en marche

3.1 Vérifications préalables

Avant toute mise sous tension, vérifier les points suivants :

- S'assurer que la pince est en position ouverte (position de repos).
- Vérifier que les câbles d'alimentation des moteurs KTECH et du Dynamixel sont correctement branchés.
- Contrôler que les deux capteurs à ultrasons HC-SR04 sont bien fixés sur leurs supports.
- S'assurer que la batterie est chargée et correctement maintenue dans son support.
- Vérifier que la carte Arduino est connectée et que le câblage bus CAN est en place.

3.2 Démarrage

- Placer le robot dans l'arène, orienté face au mur DC (mur de droite).

- Connecter la batterie au circuit d'alimentation.
- Appuyer sur le bouton lumineux vert pour lancer le programme principal.
- Le robot exécute automatiquement la séquence d'initialisation (calibrage).

⚠ Ne pas déplacer le robot après avoir appuyé sur le bouton : le calibrage se lance immédiatement.

4. Fonctionnement autonome

Une fois démarré, le robot exécute de façon autonome la séquence suivante :

4.1 Calibrage initial

Fonction : `calibrage_initial()`

Le robot se place à 30 cm du mur DC afin de se positionner au même niveau que le totem. La procédure est la suivante :

- Rotation vers la droite.
- Avancée jusqu'à atteindre 30 cm du mur (mesure par capteur ultrason).
- Rotation vers la gauche pour retrouver son orientation initiale.

Remarque: Cette étape est essentielle : si le robot démarre trop loin ou trop près du mur, le calibrage peut échouer.

4.2 Recherche de l'objet

Fonction : `chercher_objet()`

Le robot se déplace dans l'arène et utilise ses capteurs à ultrasons pour détecter le totem. La détection peut se faire :

- Devant le robot (capteur frontal).
- Sur le côté droit (capteur latéral).

Pendant cette phase, un correcteur de trajectoire (`corriger_droite()`) maintient le robot à distance constante du mur de droite ($\pm 10\%$) pour compenser les écarts mécaniques des moteurs. Afin de trouver l'objet le capteur avant est utilisé pour déterminer à quelle distance le robot se trouve de l'objet.

4.3 Saisie de l'objet

Fonction : `pince.cpp` — `pince()`

Lorsque le totem est détecté à la bonne distance, le robot lève le moteur Dynamixel pour prendre l'objet.

4.4 Déplacement vers la zone de dépôt

Fonction : retour() — sous-fonctions : degtorad2()

Le robot se dirige vers la zone de dépôt en utilisant la combinaison de deux fonctions :

- Tourner (angle, vitesse) : fait pivoter le robot de l'angle demandé.
- Avancer (distance, vitesse) : fait avancer le robot de la distance demandée.

La correction de trajectoire reste active pendant tout le déplacement.

4.5 Dépôt de l'objet

Fonction : retour.cpp — retour()

Une fois en zone de dépôt, le robot ouvre la pince pour relâcher l'objet. La pose est automatique.

6. Dépannage courant

Le robot ne démarre pas

- Vérifier que la batterie est chargée et correctement branchée.
- S'assurer que le bouton lumineux vert est bien soudé et fonctionnel.
- Contrôler les connexions des moteurs sur le bus CAN.

Le robot ne détecte pas l'objet

- Vérifier que les capteurs HC-SR04 sont bien orientés et fixés sur leurs supports.
- S'assurer que le totem se trouve bien dans la zone de détection (devant ou à droite).
- Relancer depuis la position de départ en respectant l'orientation initiale.

La trajectoire est déviée

- Vérifier que les roues sont bien montées et serrées (présence de la tige de maintien).
- S'assurer que le capteur latéral n'est pas obstrué.
- Vérifier l'alimentation des moteurs KTECH.

La pince ne se ferme pas / ne s'ouvre pas

- Vérifier le câblage du moteur Dynamixel.
- Contrôler l'alimentation dédiée au Dynamixel (câble soudé séparément).

Code ou câblage défaillant

Remarque : En cas de dysfonctionnement grave, procéder à une réinitialisation complète du code et du câblage.

7. Entretien et précautions

- Ne pas laisser la batterie branchée lorsque le robot est inutilisé.
- Vérifier l'état des soudures des câbles avant chaque utilisation (câbles d'alimentation, bouton).
- Contrôler l'intégrité du châssis MDF : le bois peut se fissurer en cas de choc.
- Les pièces imprimées en 3D (support pince, maintien batterie) peuvent se déformer sous chaleur : ne pas laisser le robot exposé au soleil.
- Après chaque utilisation, remettre la pince en position ouverte pour éviter la fatigue du servomoteur.

Revision #45

Created 12 February 2026 09:44:25 by Antoun Elias

Updated 28 May 2026 08:21:00 by Zeddama Aya