

Tableau de bord - CNChess

Période du sprint : 28 janvier au 3 février

Objectif du projet : CNChess a pour objectif de créer un adversaire automatisé pour un jeu d'échec. Le système est capable de planifier ses prochains coups, de déplacer ses pièces de manière autonome et d'afficher en temps réel l'état de la partie à l'écran.

Membres de l'équipe

Julien St-Jean (8h)

Céleste-Lyi Martel (8h)

Emanuel Marcotte (10h)

Christophe Laflamme (8h)

Zacharie Roy (5h)

Thomas Lepage-Levesque (8h)



Électronique

Électronique

Mécanique

Mécanique

Programmation

Programmation

Sous-équipe	Fait	À faire
Mécanique	<ul style="list-style-type: none"> • CAO du plateau + chute des pièces • CAO du core XY • CAO de l'effecteur 	<ul style="list-style-type: none"> • Finaliser la chute de pièces • Intégration des limit switches et des cable chains dans le CAO • Début du prototypage
Électrique	<ul style="list-style-type: none"> • Première version du circuit et tests sur breadboard • Dessin du schéma électrique • Commande des composantes 	<ul style="list-style-type: none"> • Tests finaux sur breadboard avec toutes les composantes • Conception du PCB
Programmation embarquée	<ul style="list-style-type: none"> • Mise en équation du core XY 	<ul style="list-style-type: none"> • Communication avec l'ESP-32 • Programmation des moteurs
Programmation logicielle	<ul style="list-style-type: none"> • Implémentation des règles de jeu et de l'affichage du plateau dans l'interface utilisateur • Path finding du déplacement des pièces 	<ul style="list-style-type: none"> • Docker avec un UI fonctionnel • Visualisation du path finding dans le terminal
Problèmes rencontrés		Solution
Commandé le mauvais port barrel jack		Commander à nouveau pour avoir le bon
Problème avec l'utilisation de Stockfish qui est local à chaque utilisateur		Utiliser Docker
Moteur Dynamixel difficile à implémenter		Utiliser un servo 9 g plus simple

