

# 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)



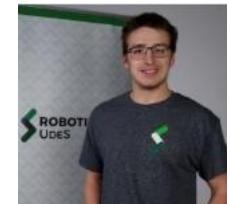
Christophe Laflamme (8h)



Zacharie Roy (5h)



Thomas Lepage-Levesque (8h)



Électronique

Électronique

Mécanique

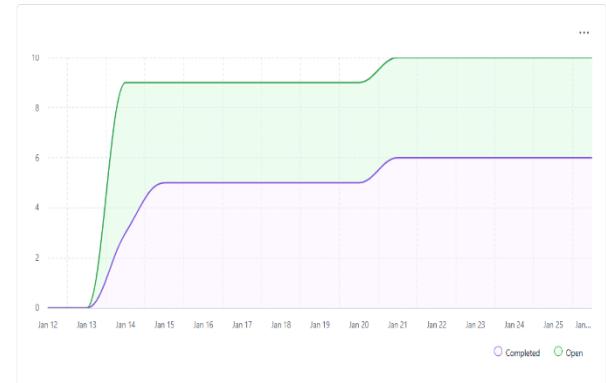
Mécanique

Programmation

Programmation

Sous-équipe	Fait	À faire
Mécanique	<ul style="list-style-type: none"> <li>• CAO du plateau + chute des pièces</li> <li>• CAO du core XY</li> <li>• CAO de l'effecteur</li> </ul>	<ul style="list-style-type: none"> <li>• Finaliser la chute de pièces</li> <li>• Intégration des limit switches et des cable chains dans le CAO</li> <li>• Début du prototypage</li> </ul>
Électrique	<ul style="list-style-type: none"> <li>• Première version du circuit et tests sur breadboard</li> <li>• Dessin du schéma électrique</li> <li>• Commande des composantes</li> </ul>	<ul style="list-style-type: none"> <li>• Tests finaux sur breadboard avec toutes les composantes</li> <li>• Conception du PCB</li> </ul>
Programmation embarquée	<ul style="list-style-type: none"> <li>• Mise en équation du core XY</li> </ul>	<ul style="list-style-type: none"> <li>• Communication avec l'ESP-32</li> <li>• Programmation des moteurs</li> </ul>
Programmation logicielle	<ul style="list-style-type: none"> <li>• Implémentation des règles de jeu et de l'affichage du plateau dans l'interface utilisateur</li> <li>• Path finding du déplacement des pièces</li> </ul>	<ul style="list-style-type: none"> <li>• Docker avec un UI fonctionnel</li> <li>• Visualisation du path finding dans le terminal</li> </ul>
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	

Risques	Solution
Prototypage du plateau plus long que prévu	Commencer le plus rapidement possible le prototypage
Délai prévu pour la livraison des pièces pas assez grand	Avancer les autres parties du projet le plus possible



Burn up chart pour le sprint 1