



POLYTECH<sup>°</sup>  
NICE SOPHIA



UNIVERSITÉ  
CÔTE D'AZUR

PROJET ARDUINO PEIP2 2022-2023

HEILMANN HUGO

HAGHIGHI CAMILIA

JEU D'ÉCHEC AUTOMATISÉ



# SOMMAIRE

## I - INTRODUCTION

1. Objectif du projet
2. Motivations
3. Matériel et conception

## II - MONTAGES ET ALGORITHME

1. Mécanisme
2. Montage électrique
3. Représentation visuel

## III - COÛT TOTAL DU PROJET

1. Coût du matériel
2. Temps de travail

## IV - PLANNING

1. Planning initial et final
2. Comparaison

## V - PROBLÈMES ET SOLUTIONS

## CONCLUSION

1. Objectifs atteint
2. Objectifs non atteint
3. Perspectives d'avenir

## BIBLIOGRAPHIE

## I - INTRODUCTION

Notre projet consiste en un jeu d'échec automatisé : on décide de nos mouvements sur un ordinateur et les pions se déplacent "tout seuls", à l'aide d'un aimant en-dessous du plateau.

- Le joueur joue sur l'ordinateur, dans un premier temps cela se fera via le moniteur série.
- Le programme donne la position et demande aux moteurs de s'y déplacer. Arrivé à l'emplacement voulu, l'aimant "attrape" la pièce pour la déplacer sur la case indiquée.

Les objectifs de ce projet étaient principalement l'amusement, pouvoir jouer à ce jeu avec une version modernisée, pouvoir améliorer ses stratégies.

Nous avons de suite pensé que ce ce projet pouvait être très intéressant et impressionnant à mettre en place, comparé à nos cours d'électronique où c'était assez théorique, ce projet est bien plus concret, il y a un montage à réaliser, de multiples compétences à mettre en oeuvre : construction mécanique (onShape), bricolage, mécanique, informatique...

Carte ARDUINO UNO



**3 moteurs pas à pas NEMA17** : 2 pour l'axe x, 1 pour l'axe y.

En effet, nous avions besoin de moteurs très précis pour pouvoir se positionner aux positions exactes voulues, et de moteurs qui peuvent facilement faire plusieurs tours "lentement" afin de faire tourner les courroies du mécanisme.

**3 drivers A4988** : qui permettent de contrôler les moteurs pas à pas.



**Electro-aimant 12V, 25 kg** : en duo avec le relais.



**Relai** : permet de convertir le 12V de l'aimant en 5V pour le brancher à l'ARDUINO et de contrôler l'activation de l'électroaimant.



**Interrupteur de fin de course** : permettent d'initialiser la position (0,0) de l'électroaimant.

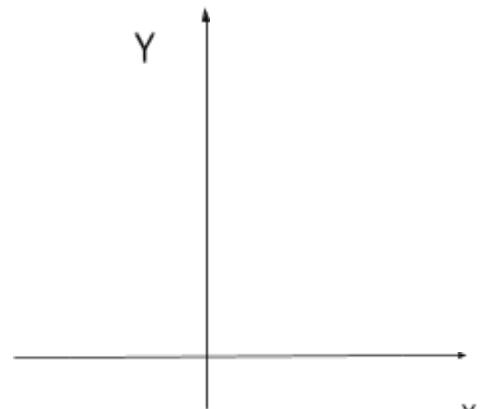
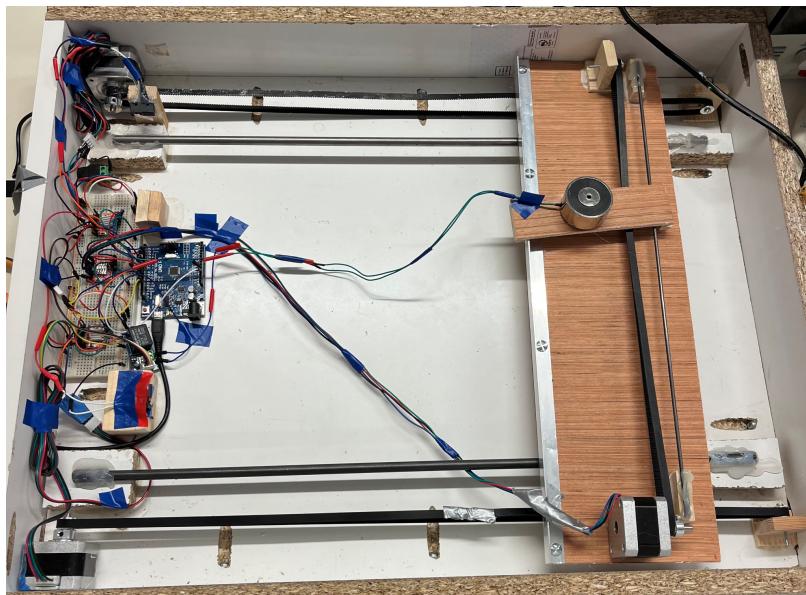


### Autre matériel:

- Des courroies pour créer le mouvement.
- Des vis.
- Des câbles.
- Une plaquette d'essai.
- Du bois (afin de créer une boîte de dimension 63x50).
- Une planche en bois que nous avons peinte.
- Une imprimante 3D.
- 32 vis que nous avons fixées sur le socle de nos pions.
- Des barres de fer

## II - MONTAGE ÉLECTRIQUE, MÉCANISME ET ALGORITHME

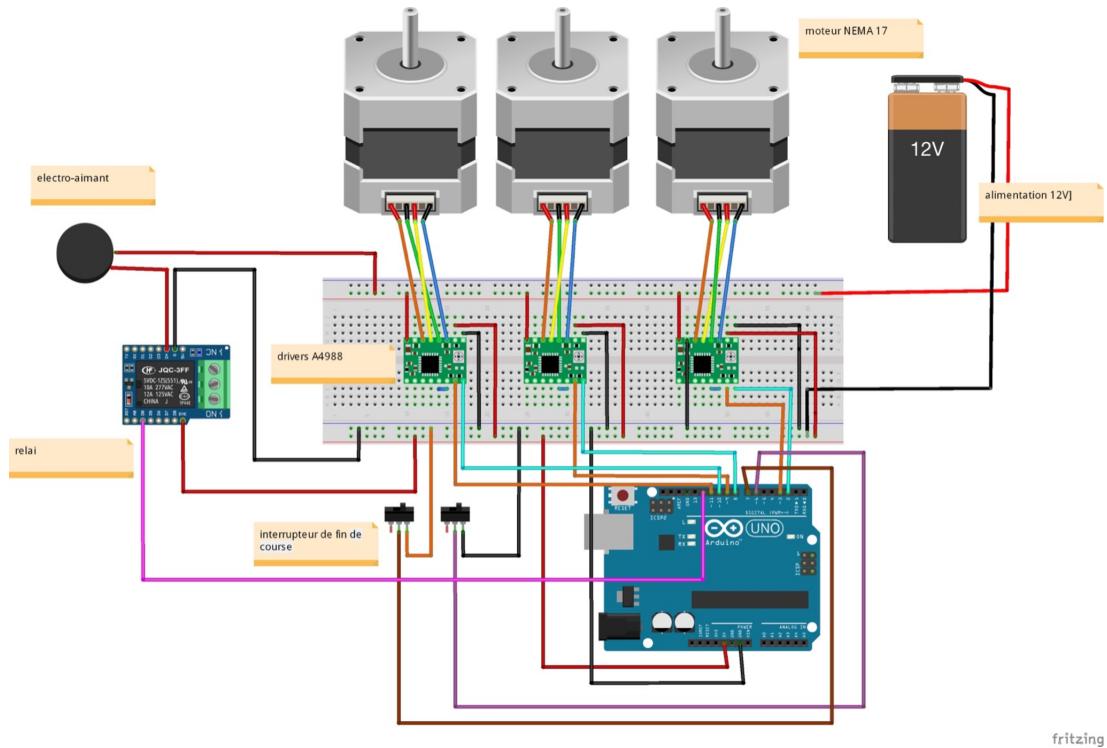
### 1. Mécanisme



Pour la structure mécanique de notre projet, nous avons utilisé deux barres de fer qui soutiennent une planche de bois en hauteur. Sur cette planche de bois, nous avons collé des courroies afin de pouvoir les entraîner selon les mouvements des moteurs. Ainsi, les barres de fer servent de support et permettent au mouvement d'être rectiligne uniforme selon l'axe des x.

Sur le même principe, nous avons mis un moteur sur cette planche qui contrôle une courroie qui est aussi fixée sur une autre planche de bois afin de créer un mouvement selon l'axe des y.

## 2. Montage électrique

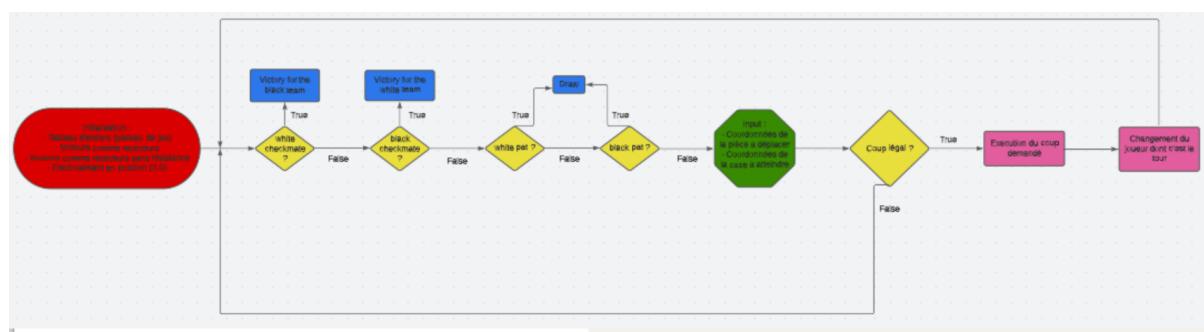


Le montage électrique de notre projet est très simple, car répétitif, car nous avons peu de composantes et certaines en double ou triple.

Il faut cependant faire attention, car nous avons sur la plaque d'essai (breadboard) un côté en 12V et un autre connecté à l'ARDUINO en 5V.

Ainsi, il est facile de se tromper et d'abîmer les composants et/ou notre matériel. (ordinateurs notamment)

## 3. Algorithme



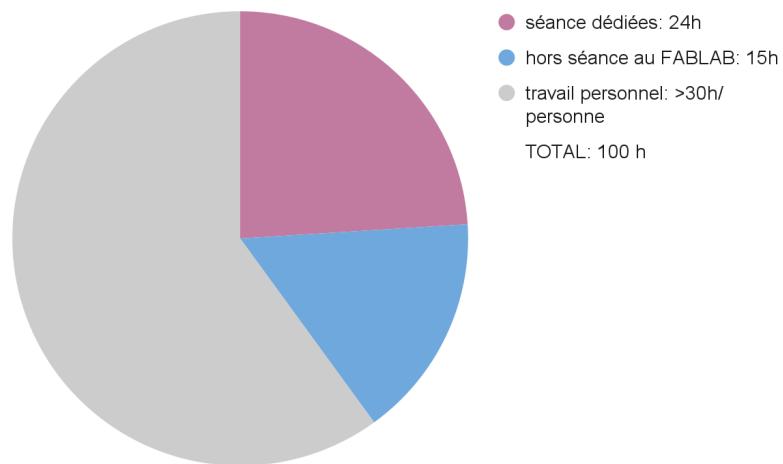
L'algorithme permettant au joueur de jouer est en soi assez basique. Tout d'abord, le programme s'initialise en déplaçant l'électroaimant en butée des deux interrupteurs de fin de course afin d'initialiser la position (0,0). Ensuite, un tableau d'entiers représentant la partie en cours est généré.

Suite à cela, le jeu peut commencer. À chaque tour, le programme va déclarer, si les conditions sont réunies, un gagnant s'il y a échec et mat ou un match nul s'il y a pat. Ensuite, le programme va demander au joueur les coordonnées de la pièce à déplacer, et ceux de la case à atteindre. Si le coup est légal, le programme exécute le mouvement en déplaçant la pièce grâce à l'électroaimant puis actualise le tableau d'entiers afin qu'il soit toujours au courant du positionnement de chaque pièce. À la fin du tour d'un joueur, le booléen "white\_turn" (initialisé à "true") change afin que le programme sache quel est le joueur qui devra jouer au prochain tour.

Ainsi, les tours vont s'enchaîner de cette manière jusqu'à ce qu'un gagnant ou un match nul soit déclaré.

### III - COÛT TOTAL DU PROJET

Matériel	Quantité	Coût moyen (en €)
Moteur NEMA 17	3	18 (l'unité)
drivers A4988	3	10 (les 5)
carte ARDUINO UNO	1	20
electro-aimant	1	16
relai	1	3
interrupteur de fin de course	2	2 (les 2)
cables, bois, vis, fil pour l'imprimante 3D		environ 20
<b>TOTAL</b>		<b>121</b>



Sachant que le salaire brut annuel d'un ingénieur est d'environ 38k euros pour 1600h de travail, on estime le prix de notre main d'œuvre à environ 2375 €. Ainsi, le coût total de notre projet est estimé à **2496€**.

## IV - PLANNING

### planning initial

	SEANCE 1	SEANCE 2	SEANCE 3	SEANCE 4	SEANCE 5	SEANCE 6	SEANCE 7	SEANCE 8	SEANCE 9
Choix du matériel Conception des pions Mise en place du système mécanique									
Impression des pions Réalisation de la maquette du système mécanique Test de l'électro-aimant avec les pions									
Réalisation du code pour le jeu d'échecs (les règles du jeu, vérifier si la case est déjà occupée, la réponse de l'ordinateur..),									
Codage des coordonnées (nombre de rotation pour se déplacer d'une colonne/ ligne, pour de déplacer d'une case à l'autre etc..)									
Programme pour le moteur (le moteur sait où il se trouve et sait où il doit aller) Programme pour l'aimant (pouvoir demander à l'aimant de s'activer / désactiver au moment voulu)									
Debug, améliorations des codes, test du moteur et de l'aimant, est ce que tout marche parfaitement?									
Interface graphique sur l'ordinateur où l'on peut visualiser le jeux d'échecs en temps réel (si l'avancement du projet le permet)									
Soutenance du projet									

### planning final:

	SEANCE 1	SEANCE 2	SEANCE 3	SEANCE 4	SEANCE 5	SEANCE 6	SEANCE 7	SEANCE 8	SEANCE 9	HORS SEANCE	OBJECTIF NON ATTEINT
Choix du matériel Conception des pions Choix du mécanisme											
Réalisation de la maquette du système mécanique											
Réalisation des pions (impression, aimant..)											
Réalisation du code pour le jeu d'échecs (les règles du jeu, vérifier si la case est déjà occupée, la réponse de l'ordinateur..),											
Code du moteur (initialisation, mouvements en fonction des règles du jeu) Code de l'électro-aimant											
Montage électrique (moteur, interrupteur, aimant...)											
Debug, améliorations des codes, test du moteur et de l'aimant, Test de l'électro-aimant avec les pions, est ce que tout marche parfaitement?											
Interface graphique sur l'ordinateur où l'on peut visualiser le jeux d'échecs en temps réel											
Soutenance du projet											

Nous n'avons aucunement respecté notre planning fixé au départ, nous supposons que nous n'avions pas prévu tous les obstacles que nous allions rencontrer et sous-estimer le temps nécessaire à certaines tâches.

On s'est beaucoup éparpillé en essayant de tout faire, mais beaucoup de choses ont été abandonnés en cours de route. (ex: interface graphique, meilleur code...)

Cependant, nous avions bien ciblé les tâches que nous allions rencontrer.

## V - PROBLÈMES ET SOLUTIONS

Nous avons rencontré beaucoup de problèmes lors de la conception de ce projet.

Notamment du point de vue matériel, en effet, nous avions commandé un électroaimant de 8kg lors des premières séances, lorsque nous l'avons reçu, nous nous sommes rendu compte que les aimant que nous avions utilisé au départ pour les pièces s'attirent entre elle et faisaient n'importe quoi, ainsi nous avons changer ces aimants par un métal plus doux comme celui d'une vis.

On a constaté que ce n'était pas du tout assez puissant pour attirer des pièces à travers une planche de bois de 3cm donc nous avons commandé un nouvel électro-aimant, cette fois-ci, de 25kg.

Malheureusement, ce problème a persisté et nous nous sommes donc retournés vers les pièces, nous avons dû redimensionner et réimprimer en 3D nos pièces pour les agrandir afin de pouvoir mettre une tête de vis plus épaisse qui permet une bonne accroche avec l'électro-aimant à travers la planche.

Lorsque nous avons réalisé notre montage pour la première fois, nous avions seulement un moteur pour contrôler le côté en x, car nous pensions que cela était suffisant. Nous nous sommes aperçu que pas vraiment ainsi nous avons dû ajouter un troisième moteur à notre système. Mais à ce moment-là, les moteurs décident de plus marcher. Nous avons passé plusieurs heures, avec l'aide de notre professeur, à chercher le problème, changer de driver, changer de moteur...

Le problème venait du câblage en effet lorsqu'on commence à avoir beaucoup de composants sur la même plaquette, cela peut créer des interférences et/ou bruits parasites. On a donc dû débrancher, rebrancher la totalité des câbles 3,4 fois et ressoudre plusieurs fils.

Un autre problème que nous avons rencontré se situe au niveau du code et notamment à cause de nos non-connaissances en C/C++.

En effet, nous voulions créer un tableau de type "Pièces" afin que le codage des règles de jeu soit plus simple. Or, la classe "Pièces" n'existant pas, il fallait la créer, ainsi que les classes "Pion", "Cavalier", "Fou", etc... avec un système d'héritage entre les classes. Nous avons donc eu beaucoup de mal pour coder une classe un minimum décente et cela s'est soldé par un échec, car il y avait beaucoup d'erreurs que nous étions incapable de comprendre. Nous avons donc fini par abandonner et retourner sur l'idée de départ de faire un tableau de nombre entiers malgré que ce soit plus difficile et plus long à coder.

Finalement, on s'est également rendu compte que notre système n'était peut-être pas le plus optimal, c'est-à-dire, que nos planches n'avaient pas parfaitement car, elle ne tenait pas droit ce qui bloquait tout le système. Si c'était à refaire, nous aurions fait un système à bille qui coulisse mieux et de façon assistée.

## CONCLUSION

En résumé, nous avons créé une boîte de toute pièce permettant de cacher tout le système mécanique et électronique que nous avons construit.

Idéalement, le système mécanique devait être fait de la manière suivante : deux moteurs reliés à des courroies font progresser une fine plaque de bois selon l'axe X. Sur cette plaque, un troisième moteur doit faire progresser une petite plaque de bois selon l'axe Y. Sur cette petite plaque se trouve l'électroaimant qui doit être déplacé sous la case de la pièce à déplacer, être activé pour "attraper" la pièce, la déplacer jusqu'à la case à atteindre, puis la lâcher.

Malheureusement, bien que l'électroaimant puisse "attraper" les pièces et les déplacer et que l'initialisation de la position (0,0) fonctionne, les deux plaques de bois ne se déplacent pas de manière rectiligne et uniforme; cela a pour conséquence que la plaque part de travers, frotte contre les parois de la boîte, donc la position finale de l'électroaimant n'est pas celle escomptée.

En ce qui concerne le code, il y a des problèmes au moment où le joueur doit renseigner les coordonnées de la pièce à déplacer et de la case à atteindre. En effet, le code est censé créer une boucle d'attente infinie le temps que le joueur renseigne les informations, or celle-ci ne fonctionne pas.

D'un autre côté, il y a des erreurs dans les méthodes permettant de savoir si un coup est légal, ce qui fait que même si la partie mécanique fonctionnait, le jeu serait injouable.

Si nous avions eu 9 séances de plus, nous aurions changé notre système afin que le mouvement soit parfaitement rectiligne et uniforme. Peut-être à l'aide d'un système de guidage à billes... Ou tout simplement essayer d'aligner correctement les courroies pour que tout soit parallèle.

En ce qui concerne le code, nous aurons pu arranger les erreurs des boucles infinies et des méthodes pour les coups légaux.

De plus, une fois tous ces problèmes réglés, nous aurions pu revenir sur notre idée de départ de créer une application/ interface graphique. Nous nous sommes penché dessus lors du projet mais par manque de temps et également dû aux lacunes en langage C/C++ nous avons abandonné.

## BIBLIOGRAPHIE

github de projet similaires qui nous ont aidés:

<https://projecthub.arduino.cc/Moustafaidris97/06bbd228-e0fc-43ee-bfae-95b3cea732fd>

<https://projecthub.arduino.cc/maguerero/67db6fe9-5fdc-44ed-9be3-9c599cc61003>

[https://projecthub.arduino.cc/carlos\\_pendas/971f8f91-f663-4903-904f-0816ddf5bb28](https://projecthub.arduino.cc/carlos_pendas/971f8f91-f663-4903-904f-0816ddf5bb28)