

A black and white photograph of chess pieces on a board is positioned on the left side of the cover. It shows a king piece standing upright and a pawn piece lying on its side, suggesting a game in progress or a strategic move.

# jeux d'échec



REALISE PAR :  
TAGMOUTI Ghita  
CHAOUI Marwa

ENCADRE PAR :  
Mr . Mehdi Iraqui Houssaini

# sommaire

- 1 Introduction
- 2 Cahier des charges
- 3 Explication du code
- 4 Difficultés et solutions apportés
- 5 Conclusion et perspectives



# INTRODUCTION

La pratique des échecs contribue à la construction de la personnalité en encourageant l'attention, l'imagination, l'anticipation, le jugement et la confiance en soi. Le jeu d'échecs participe à développer l'apprentissage des règles et le respect d'autrui, et à ce titre participe de l'apprentissage de la citoyenneté.



# cahier des charges

Dans le cadre de ce projet , il nous a été fournie un cahier de charges a suivre comportant 4 phases:

Phase 1 : Conception et développement de la logique du programme suivant les regles universel du jeux et d'une interface utilisateur console/clavier.

Phase 2 : Conception et développement des règlesavancées.

Phase 3 : Conception et développementd'une interface graphique.

Phase 4 :implémenter les fins de partie implémenter une intelligence artificielle permettre à tout moment d'enregistrer l'état de la partie dans un fichier

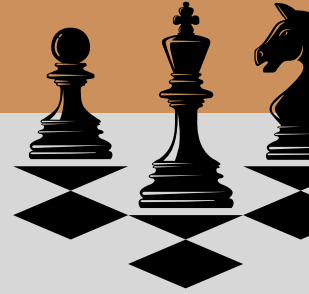




# Explication du code

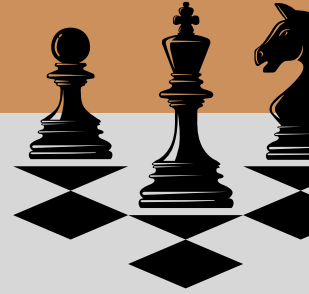
## 1. Mouvements des pièces :

- `int getIndice(int,int);` renvoie simplement la valeur de l'élément `t[i][j]`
- `int movpion(int ld,int cd,int la,int ca);`  
déplacement du pion
- `int firstmoveN(int ld,int cd,int la,int ca);`  
premier pas du pion noir
- `int firstmoveB(int ld,int cd,int la,int ca);`  
premier déplacement du pion blanc
- `int chlibretour(int ld,int cd,int la,int ca);`  
chemin libre horizontale et verticale
- `int chlibrefou(int ld, int cd, int la, int ca);`  
chemin libre en diagonal
- `int chlibredame(int ld, int cd, int la, int ca);`  
chemin libre orthogonal et horizontal
- `int movtour(int ld,int cd,int la,int ca);`  
déplacement de la tour
- `int movcavalier(int ld,int cd,int la,int ca);`  
déplacement du cavalier
- `int movfou(int ld,int cd,int la,int ca);`  
déplacement du fou
- `int movdame(int ld,int cd,int la,int ca);`  
déplacement de la dame
- `int movroi(int ld,int cd,int la,int ca);`  
déplacement du roi



## 2.Mouvements spéciaux:

- void Promotion(int ld, int cd, int la, int ca);  
consiste à le remplacer immédiatement par une autre pièce lorsqu'il arrive sur la dernière rangée
- void RookNoir(int ld , int cd , int la , int ca);  
c'est un déplacement speciale du roi noir et d'une des tours d'echecs
- void RookBlanc(int ld , int cd ,int la,int ca);  
c'est un déplacement speciale du roi noir et d'une des tours d'echecs
- void PriseEnPassant(int ld , int cd ,int la,int ca);  
c' est une possibilité particulière de capturer un pion
- int deroulement(int ld,int cd,int la,int ca);  
détermine le type de pièce présente dans la case de départ en vérifiant la valeur de t[ld][cd], puis appelle la fonction correspondante pour effectuer le mouvement de cette pièce



### 3. Continuer la partie precedente:

- Méthode save :

```
void Plateau::Save(int joueur)
```

Enregistre l'état actuel du plateau de jeu dans un fichier texte, y compris les positions des pièces et le joueur actuel

- Méthode load :

```
void Plateau::Load(int &joueur)
```

Charge l'état du plateau de jeu à partir du fichier "chess.txt", y compris les positions des pièces et le joueur actuel

### 4. Fin de partie

- `bool CheckRoiBlanc(int line , int colonne);`

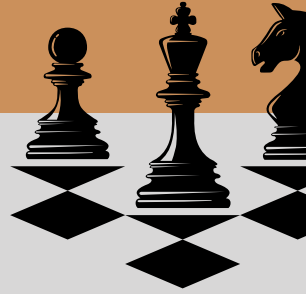
Voit si le roi blanc est en echec

- `bool CheckRoiNoir(int line , int colonne);`

Voit si le roi noir est en echec

- `int nbCoupsPossibles(int joueur);`

Calcule le nombre de coups possibles pour un joueur donné



- `bool Null();`

vérifie s'il y a suffisamment de pièces sur le plateau pour continuer la partie en comptant le nombre de pièces , il doit etre egale a 1 blanche et 1 noir

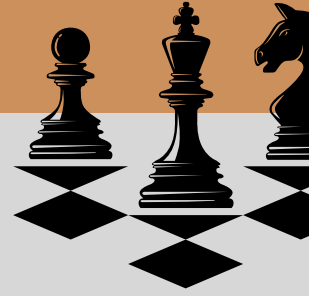
- `bool Mate();`

vérifie si le roi blanc est en échec et n'a aucun coup possible ou si le roi noir est en échec et n'a aucun coup possible et renvoie true dans ces cas. Sinon, elle renvoie false.

- `bool Finpartie();`

vérifie si la partie est terminée en appelant la fonction `Mate()` pour vérifier s'il y a un échec et mat, et la fonction `Null()` pour vérifier si la partie est nulle





## 5.Intelligence Artificielle

- `int nbPiecesIA();`

renvoie le nombre de pièces contrôlées par l'IA sur le plateau de jeu.

- `int nbPiecesAdversaire();`

compte le nombre de pièces contrôlées par l'adversaire de l'IA

- `static void jouerCoup(Plateau& plateau, int joueur);`

implémente la logique pour déterminer le meilleur coup pour le joueur IA en utilisant les méthodes fournies par la classe Plateau et affiche des messages appropriés en cas d'événements spécifiques tels que le roi en échec

# Difficultés rencontrées

Lors du développement du jeu d'échecs, on a rencontré plusieurs difficultés. Premièrement, la mise en place de la logique des mouvements et des règles spécifiques des pièces s'est révélée complexe, nécessitant une compréhension approfondie du jeu. Deuxièmement, la gestion des différentes conditions de fin de partie, telles que le mat, et le match nul a posé des défis supplémentaires pour garantir un déroulement correct du jeu. Enfin, l'implémentation de l'intelligence artificielle pour prendre des décisions stratégiques et jouer contre l'adversaire a nécessité une approche réfléchie et un équilibre entre performance et jouabilité.



## SOLUTIONS

Pour surmonter les difficultés rencontrées, on a utilisé des ressources en ligne telles que des forums de développeurs et des tutoriels pour trouver des solutions à des problèmes spécifiques. On a également pris le temps de décomposer les problèmes complexes en tâches plus petites ce qui nous a permis de résoudre les difficultés étape par étape.



# REMERCEMENT

Nous tenons à exprimer notre sincère gratitude envers notre précieux professeur de programmation, M. Iraqui, pour son encadrement précieux tout au long de ce projet. Sa présence et ses conseils avisés ont grandement contribué à notre apprentissage dans le domaine de la programmation et à notre compréhension approfondie du jeu d'échecs. Nous lui sommes profondément reconnaissants pour son dévouement et sa passion, et nous le remercions chaleureusement pour son soutien continu. Merci, M. Iraqui.