

Stackchess



Lucas HERFORT groupe 2.1

Sommaire

I/ Les objectifs du projet.....	Page 3
II/ Mise en place du projet.....	Page 7
III/ Interface du projet.....	Page 9
IV/ Déplacement des pièces.....	Page 14
V/ Déplacement du pion.....	Page 15
VI/ Déplacement de la tour.....	Page 20
VII/ Déplacement du fou.....	Page 23
VIII/ Déplacement du roi et de la dame.....	Page 26
IX/ Déplacement du cavalier.....	Page 30
X/ Utilisation et mise en place des piles.....	Page 32
XI/ Amélioration et correction des bugs.....	Page 39
XI/ Conclusion et limite du programme.....	Page 45

I/ Les objectifs du projet

Le but de ce projet est d'implanter un jeu d'échecs modifiés, *Stackchess*.

Dans cette version modifiée du jeu d'échecs, il est possible d'empiler les pièces d'une même couleur sur les cases. À chaque tour, un joueur peut déplacer un nombre quelconque de pièces situées sur le haut de la pile d'une même case. Néanmoins, le déplacement doit être légal pour chacune de pièces déplacées. Par ailleurs, il ne suffit pas de se déplacer sur une case contenant une pièce adverse pour la prendre ; il faut déplacer un nombre de pièces strictement supérieur à celui des pièces adverses dans la case atteinte.

Le but du jeu n'est plus de prendre le roi de l'adversaire, mais de prendre toutes ses pièces. En particulier, il ne sera pas obligatoire pour un joueur de sortir d'une situation d'échec (cas où le roi du joueur peut être pris le tour suivant).

Fonctionnement

On joue sur un plateau de taille $n \times n$. Par défaut, n vaudra 6, mais une extension décrite ci-dessous indique comment généraliser ceci.

On utilisera les pièces suivantes, dont on rappelle les mouvements légaux :

Roi (notation R)

peut bouger d'une case verticalement, horizontalement ou en diagonale.

Dame (notation D)

peut bouger d'un nombre quelconque de cases verticalement, horizontalement ou en diagonale.

Fou (notation F)

peut bouger d'un nombre quelconque de cases en diagonale.

Tour (notation T)

peut bouger d'un nombre quelconque de cases verticalement ou horizontalement.

Pion (notation P)

- si la case visée ne contient pas de pièce adverse : peut bouger d'une case en direction de l'adversaire, de deux si le pion n'a pas encore bougé de la partie.
- si la case visée contient au moins une pièce adverse : peut bouger d'une case en diagonale en direction de l'adversaire.

On ne peut pas sauter par dessus une pièce (mais on peut se placer dessus si elle est de la même couleur).

NB: pour simplifier on ne considérera pas les cavaliers dans un premier temps, étant donné que seul un cavalier peut avoir les mêmes déplacements qu'un cavalier. De même, on n'autorisera pas les roques et les prises en passant.

- Initialement, le plateau est le suivant, où le nom d'une pièce est suivi de sa couleur :

	a	b	c	d	e	f
	+	-	-	+	-	-
6	TN	FN	DN	RN	FN	TN
	+	-	-	+	-	-
5	PN	PN	PN	PN	PN	PN
	+	-	-	+	-	-
4						
	+	-	-	+	-	-
3						
	+	-	-	+	-	-
2	PB	PB	PB	PB	PB	PB
	+	-	-	+	-	-
1	TB	FB	DB	RB	FB	TB
	+	-	-	+	-	-

- Chaque joueur joue à tour de rôle, en commençant par les blancs.
- À son tour, un joueur choisit une case et déplace un nombre qu'il choisit de pièces en haut de la pile sur cette case. Le déplacement n'est possible que si le coup est légal pour toutes les pièces déplacées. Les pièces conservent leur ordre. En particulier, si la case cible est occupée par des pièces du joueur, les pièces s'ajoutent dans le même ordre au-dessus des pièces déjà présentes. Si la case cible est occupée par des pièces de l'adversaire, le déplacement n'est possible que si on déplace strictement plus de pièces qu'il n'y a des pièces de l'adversaire sur la case cible. Dans ce cas, les pièces de l'adversaire sont prises et retirées du jeu.
- La partie se termine quand un des joueurs a pris toutes les pièces de son adversaire, ou si un des joueurs abandonne. Il peut également y avoir un cas d'égalité quand chacun des joueurs ne possède que des pièces distinctes qui n'ont pas de déplacement légal en commun, par exemple un fou et une tour. (En effet, dans ce cas, il est impossible de prendre les pièces de l'adversaire puisqu'il n'est plus possible de déplacer deux pièces en même temps.)

Interface

On travaille dans un premier temps en mode texte. L'affichage se compose de deux parties : à gauche, le plateau de jeu vu de dessus (on ne voit donc que les pièces au sommet des piles) ; à droite, les coordonnées et le contenu de la pile de pièces sur une case sélectionnée.

(Dans un premier temps, on pourra se contenter d'afficher cette pile horizontalement et sous le plateau de jeu.)

Pour afficher les pièces au sommet de la pile, on affichera comme dans l'exemple ci-dessus le nom de la pièce et sa couleur.

En dessous du tableau se trouve une ligne indiquant de quel joueur c'est le tour, et un invite de choix d'action.

À son tour, un joueur entre les coordonnées d'une case (par exemple c5). L'affichage se met à jour en mettant à droite le contenu de la case qui vient d'être sélectionnée. Le

joueur peut alors entrer une lettre pour effectuer les actions suivantes :

- avec c on change la case sélectionnée. Il faut alors entrer de nouvelles coordonnées, l'affichage est remis à jour et le joueur recommence à entrer une lettre.
- avec d on déplace des pièces depuis la case sélectionnée. On vérifie qu'il y a au moins une pièce sur cette case et qu'elle appartient bien au joueur dont c'est le tour. Si c'est le cas, le joueur entre un nombre de pièces à déplacer ; ce nombre doit être compris entre 1 et le nombre de pièces sur la case. Le joueur entre ensuite la coordonnée cible. Si le déplacement est légal, il est effectué, et les pièces adverses sont prises le cas échéant. Si le déplacement n'est pas possible, un message d'erreur informatif est affiché et le joueur recommence à entrer une lettre.
- avec a le joueur abandonne la partie.

L'affichage est rafraîchi après chaque déplacement, et on change de joueur.

À chaque fois qu'on change de joueur, il faut vérifier que la partie n'est pas terminée (prise de toutes les pièces de l'adversaire, ou égalité).

Exemple d'affichage attendu:

```

a  b  c  d  e  f
+--+--+--+--+--+--+
6 | TN | FN | DN | RN | FN |   |
+--+--+--+--+--+--+
5 | PN | PN | PN | PN | PN |   |
+--+--+--+--+--+--+
4 |   |   |   |   |   |   |
+--+--+--+--+--+--+
3 |   |   |   |   |   | TN |
+--+--+--+--+--+--+
2 | TB | PB | PB | PB | PB | PB |
+--+--+--+--+--+--+          TB
1 |   |   | DB | RB | FB | TB |          FB
+--+--+--+--+--+--+      a2: PB

```

C'est au tour des blancs. Quel est votre choix ?

Extensions

Plateau de taille quelconque

La taille n du plateau sera demandé en début de partie. Pour la disposition initiale, on mettra la dame et le roi au milieu, et on alternera autour des séquences fou puis tour.

Exemple pour $n=9$:

	a	b	c	d	e	f	g	h	i
9	FN	TN	FN	DN	RN	FN	TN	FN	TN
8	PN	PN	PN	PN	PN	PN	PN	PN	PN
7									
6									
5									
4									
3									
2	PB	PB	PB	PB	PB	PB	PB	PB	PB
1	FB	TB	FB	DB	RB	FB	TB	FB	TB

Dames

Quand des pièces arrivent sur la dernière rangée, les pions arrivés deviennent des dames.

Cavaliers

On introduit les cavaliers (notés C). Les cavaliers se déplacent en L: une case verticalement et deux cases horizontalement, ou deux cases verticalement et une case horizontalement. Dans cette extension, leur particularité est qu'ils peuvent « porter » une autre pièce : si on déplace deux pièces dont un cavalier, alors les deux pièces pourront se déplacer en L quelle que soit l'autre. Plus généralement, on peut déplacer en L un nombre k inférieur ou égal à $2m$ pièces si parmi ces k pièces se trouvent m cavaliers.

Pour remplir les plateaux de taille quelconque, on alternera autour de la dame et du roi des séquences de fou, puis cavalier, puis tour.

Remarque : si un pion se déplace avec un cavalier et retourne sur la deuxième ligne, il ne pourra toutefois pas avancer de deux cases un tour ultérieur. En effet, ce mouvement n'est possible que pour son premier déplacement.

II/ Mise en place du projet

Les fonctions utilisées pour réaliser le projet sont :

Nom de la fonction	Description
stack create()	Créer et retourne une pile vide.
int is_empty (stack s)	Retourne 1 si la pile s est vide et 0 sinon.
void push (char* i, stack *s)	Ajoute l'élément i au sommet de la pile s.
char* pop(stack* s)	Supprime le sommet de la pile s.
char* sommet(stack s)	Retourne le sommet de la pile s.
int TaillePile(stack s)	Retourne la taille de la pile s.
void Initialisation ()	Elle permet : l'affichage du plateau de jeu, la création et empilement de toutes les piles du l'échiquier à l'état initial. Cette fonction n'est appelée qu'une seule fois.
void Game ()	Elle permet : la terminaison de la partie, la gestion des joueurs, le choix de la case de départ sur l'échiquier.
void Déplacement (int i)	Si l'entier i passé en argument est égal à 1, la fonction renvoie la pile de départ choisie par le joueur et la variable départ qui prend le numéro de la case de départ sur l'échiquier et lance la fonction Rafraîchir. Si l'entier i passé en argument est différent de 1, la fonction renvoie la pile d'arrivée choisie par le joueur et la variable arrive qui prend le numéro de la case d'arrivée sur l'échiquier.
void Rafraîchir (stack p, int aff)	Comme la fonction Initialisation , elle actualise l'affichage de l'échiquier mais en plus le nom et les éléments de la pile p passé en argument. Si le paramètre aff passé en argument est égal à 0, elle demande au joueur de choisir une lettre parmi 'a' , 'c' ou 'd' qui effectue les actions définies dans les objectifs du projet. Le paramètre aff est égal à 1 lorsque le joueur en cours à fini de jouer. Dans ce cas, elle appelle la fonction Game.
void Test ()	Elle permet de gérer le nombre de pièces à déplacer ainsi que le nombre de coups restant au joueur actuel.
void MovePiece ()	Elle permet de savoir quelle pièce se trouve au sommet de la pile de départ.
void MovePion ()	Elle permet le déplacement du pion.
void MoveTour ()	Elle permet le déplacement de la tour.
void MoveFou ()	Elle permet le déplacement du fou.
void MoveDameEtRoi ()	Elle permet le déplacement de la dame et du roi en faisant appel aux fonctions MoveTour et MoveFou.
void MoveCavalier ()	Elle permet le déplacement du cavalier.
void PopAndPush ()	Elle permet d'empiler et de désempiler les piles lorsque les joueurs bougent des pièces.

Les variables utilisées pour réaliser le projet sont :

Nom de la variable	Description
struct stack {char* content[N]; int summit;}; typedef struct stack stack;	Implémentation des piles.
int Coordonnees[8][8]	Tableau à 2 dimensions permettant de numérotéer chaque case de l'échiquier.
char ChoixChiffre[8] char Choix Lettre[8]	Ces variables sont utilisées afin de savoir si la case de départ ou la case cible choisie par un joueur existe sur l'échiquier.
int CasePionBlancs[8] int CasePionNoirs[8]	Stocke les cases de départ des pions blancs et noirs pour savoir si un pion a déjà été joué.
char* Echiquier[64];	Tableau a une dimension pour qui contient le nom des cases de l'échiquier avec une lettre et un chiffre.
char *Piece[64];	Tableau à une dimension représentant l'échiquier qui sera rafraîchi puis utilisé pour l'affichage.
char *PieceNoire[6]; char *PieceBlanche[6];	Stocke toutes les pièces blanches et noirs.
char coup,cible;	Stocke respectivement la case de départ et la case cible que le joueur souhaite.
char NombreDePiece;	Stocke le nombre de pièces à déplacer choisi par le joueur sous forme de caractère.
int depart,arrive;	Stocke respectivement le numéro de la case de départ et de la case d'arrivée que le joueur souhaite.
int PieceDeplace,CoupRestant;	Stocke respectivement le nombre de pièces que le joueur souhaite déplacer ainsi que le nombre de pièces qui lui reste à déplacer s'il choisit d'en déplacer plusieurs.
int Joueur;	Si elle vaut 1 ce sont les blancs de jouer sinon si elle vaut 0, c'est au noirs de jouer.
int PieceBrestantes = 16; int PieceNrestantes = 16;	Stocke le nombre de pièces blanches et noires restantes sur l'échiquier afin de savoir quand le jeu se termine.
stack a1,,,,,,,,h8;	Déclare toutes les piles de l'échiquier.
Stack currentPile,PileCible;	Stocke respectivement le contenu de la pile de départ et de la pile d'arrivée que le joueur souhaite.
stack T[64];	Tableau à une dimension contenant toutes les piles de l'échiquier pour identifier la pile de départ et la pile d'arrivée.

III/ Interface du projet :

Pour pouvoir introduire le cavalier directement, j'ai décidé de commencer le projet avec un plateau de jeu de taille 8 x 8 comme pour de jeu d'échec classique (figure 1) . Les 64 cases seront numérotés (variable *Coordonnées[8][8]*) afin de les identifier (figure 2) :

```

      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8 | TN | CN | FN | DN | RN | FN | CN | TN |
+---+---+---+---+---+---+---+
7 | PN | PN | PN | PN | PN | PN | PN | PN |
+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2 | PB | PB | PB | PB | PB | PB | PB | PB |
+---+---+---+---+---+---+---+
1 | TB | CB | FB | DB | RB | FB | CB | TB |
+---+---+---+---+---+---+---+

```

Figure 1

```

      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8 |  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |
+---+---+---+---+---+---+---+
7 |  9 |10|11|12|13|14|15|16|
+---+---+---+---+---+---+---+
6 |17|18|19|20|21|22|23|24|
+---+---+---+---+---+---+---+
5 |25|26|27|28|29|30|31|32|
+---+---+---+---+---+---+---+
4 |33|34|35|36|37|38|39|40|
+---+---+---+---+---+---+---+
3 |41|42|43|44|45|46|47|48|
+---+---+---+---+---+---+---+
2 |49|50|51|52|53|54|55|56|
+---+---+---+---+---+---+---+
1 |57|58|59|60|61|62|63|64|
+---+---+---+---+---+---+---+

```

Figure 2

Pour afficher le plateau de jeu à l'état initial, j'ai créé une fonction d'initialisation puis j'ai utilisé la fonction `printf()` ainsi que la variable *Pièce[64]* décrite plus haut. Voici l'affichage que j'ai obtenu :

```

      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8 | TN | CN | FN | DN | RN | FN | CN | TN |
+---+---+---+---+---+---+---+
7 | PN | PN | PN | PN | PN | PN | PN | PN |
+---+---+---+---+---+---+---+
6 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2 | PB | PB | PB | PB | PB | PB | PB | PB |
+---+---+---+---+---+---+---+
1 | TB | CB | FB | DB | RB | FB | CB | TB |
+---+---+---+---+---+---+---+

```

Ensuite, il faut informer les joueurs pour savoir à quel moment ils doivent jouer. Pour cela, j'ai créé une fonction `Game ()`. En début de partie ce sont les blancs qui commencent à jouer :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?

```

A ce moment là, le joueur doit entrer une case de départ valide. Une case valide comporte d'abord la lettre puis après le chiffre. Pour récupérer cette information, j'ai utilisé la fonction `scanf()`. Si le joueur rentre une case qui n'existe pas ou qui n'a pas le bon format ou qui ne contient pas de pièce, un message d'erreur sera affiché et le joueur devra rentrer une nouvelle case de départ :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?
2
La case 2 n'existe pas.
C'est au tour des blancs. Quel est votre choix ?
n
La case n n'existe pas.
C'est au tour des blancs. Quel est votre choix ?
u8
La case u8 n'existe pas.
C'est au tour des blancs. Quel est votre choix ?

```

← 2 n'est pas une case

← n n'est pas une case

← u8 n'est pas une case

Dans le cas où la case choisie est correcte, je fais appel à la fonction déplacement (0) qui va stocker la valeur de cette case dans la variable *départ* et le contenu de la pile dans la variable *currentPile*. l'affichage doit être rafraîchi avec le contenu de la pile départ à droite du plateau de jeu. Pour cela, je fais appel à la fonction rafraîchir () en lui passant en paramètre la variable *currentPile*. Le plateau de jeu sera affiché de la même manière que la fonction initialisation.

Pour afficher le contenu de la pile, j'ai créé une variable local `char* AffichePile[8]` qui est un tableau contenant tous les éléments de `currentPile` passés en paramètre. Le contenu de `AffichePile[8]` sera ainsi rajouté aux fonctions `printf()` utilisées pour afficher le plateau de jeu :

```

      a b c d e f g h
+---+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+
a2: PB

```

Le joueur a choisi la case a2 comme case de départ

Ensuite comme l'impose les objectifs du projet, le joueur doit entrer une lettre :

- si la lettre est « a », le joueur abandonne la partie, donc on arrête le programme.
- si la lettre est « c », le joueur change de case de départ et l'affichage se remet à jour.
- si la lettre est « d », le joueur décide de déplacer des pièces depuis cette case.

```

      a b c d e f g h
+---+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+
a2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
a
Les noirs gagnent par forfait

```

Le joueur a entré la lettre « a »

```

Que souhaitez vous faire ? veuillez entrer une lettre :
c
      a b c d e f g h
+---+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Le joueur a entré la lettre « c »

```

a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :

```

Le joueur a entré la lettre « d »

```

Que souhaitez vous faire ? veuillez entrer une lettre :
v
La lettre v n'est pas correct
a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Le joueur a entré la lettre « v », comme c'est une lettre incorrecte, un message d'erreur apparaît et le joueur doit rentrer une lettre valide

En revanche, si le joueur choisit la lettre « d » et que les pièces présentes sur la case de départ ne lui appartiennent pas, il faut qu'un message d'erreur apparaisse et qu'on lui demande de choisir une autre case de départ :

```

C'est au tour des blancs. Quel est votre choix ?
d7

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Les pieces presentent ne vous appartient pas
a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Dans cette situation, c'est au tour des blancs et je choisis la case d7. Un message d'erreur s'affiche bien et je dois choisir une autre case de départ.

Si le joueur entre la lettre « d », il doit choisir un nombre de pièces à déplacer depuis cette case. Pour cela je fais appel à la fonction Test().

Cependant dans le cas présent, la case a2 ne contient qu'une seule pièce qui est un pion blanc, il n'est donc pas nécessaire de demander au joueur combien de pièces il souhaite déplacer.

Par contre, si *currentPile* a une taille supérieure ou égale à 2 on demandera au joueur de choisir un nombre de pièces. Ensuite le joueur doit choisir une case cible pour la pièce se trouvant au sommet de *currentPile*.

Pour cela, je fais appel à la fonction MovePiece () qui va me permettre de demander au joueur une case cible et de tester si le déplacement souhaité par le joueur est légal selon le type de pièce.

A partir de maintenant je vais devoir créer toutes les fonctions qui vont permettre de déplacer les différentes pièces.

IV/ Déplacement des pièces :

Avant de créer les fonctions nécessaires au mouvement des pièces il faut tout d'abord trouver un moyen d'identifier les pièces. Pour cela, j'ai utilisé la fonction `strchr()` de la librairie `string.h` qui renvoie `NULL` si un caractère ne se trouve pas dans une chaîne de caractère.

- si la chaîne de caractères au sommet de *currentPile* contient le caractère P, alors il s'agit d'un pion donc on peut lancer la fonction `MovePion ()`.
- si la chaîne de caractères au sommet de *currentPile* contient le caractère T, alors il s'agit d'une tour donc on peut lancer la fonction `MoveTour ()`.
- si la chaîne de caractères au sommet de *currentPile* contient le caractère F, alors il s'agit d'un fou donc on peut lancer la fonction `MoveFou ()`.
- si la chaîne de caractères au sommet de *currentPile* contient le caractère D ou R, alors il s'agit d'une dame ou d'un roi donc on peut lancer la fonction `MoveDameEtRoi ()`.
- si la chaîne de caractères au sommet de *currentPile* contient le caractère C, alors il s'agit d'un cavalier donc on peut lancer la fonction `MoveCavalier ()`.

Lors de la création du jeu, j'ai implémenté les déplacements des différentes pièces dans l'ordre indiqué ci-dessus. Pour chacune de ces fonctions de déplacement, mon choix a été de trouver des relations mathématiques simples entre le numéro de la case de départ et le numéro de la case d'arrivée provenant des variables *départ* et *arrive*.

Une fois que la pièce sera déplacée, je rafraîchirai l'affichage avec la fonction `rafraîchir` puis on changera de joueur en mettant à jour la variable *Joueur* avant de rappeler la fonction `Game ()` de départ.

V/ Déplacement du pion (P)

Critères de mouvements du pion :

- il peut avancer d'une case en direction de l'adversaire, de deux si le pion n'a pas encore bougé de la partie si la case visée ne contient pas de pièce adverse
- si une pièce adverse se trouve devant le pion, il peut bouger d'une case en diagonale en direction de l'adversaire.

	a	b	c	d	e	f	g	h
8								
7								
6								
5								
4								
3								
2								
1								

	a	b	c	d	e	f	g	h
8								
7								
6								
5								
4								
3								
2								
1								

Départ (b2) = 50
Arrivée (b3) = 42

Arrivée – Départ = -8

Départ (d7) = 12
Arrivée (d6) = 20

Arrivée – Départ = 8

Pour permettre aux pions de bouger, je vais utiliser les coordonnées des cases stockées dans la variable `Coordonnées[8][8]`. Les valeurs des cases de départ et d'arrivée sont stockées dans les variables *arrive* et *départ*.

Comme le plateau fait 8 cases de longueur, un mouvement d'une case correspond à une différence de 8 entre le numéro de la case de départ et d'arrivée prit en valeur absolue ou 16 si le pion n'a pas encore bougé.

Un pion n'a pas encore été joué s'il se trouve sur les case de 9 à 16 pour les pions noirs et de 49 à 56 pour les pions blancs. Ces cases sont stockées dans les variables *CasePionBlancs[8]* et *CasePionNoirs[8]*.

Si une pièce adverse se trouve devant le pion, alors ses deux déplacements légaux en diagonal devront se faire sur les cases *arrive – 1* et *arrive + 1*.

Si un joueur tente de déplacer un pion illégalement un message d'erreur s'affichera ; de même si un joueur tente de déplacer un pion qui a déjà été joué.

Voici un test pour un déplacement d'une case :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
a3
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3|PB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Déplacement du PB de a2 vers a3

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PN :
g6
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN| |PN|
+---+---+---+---+---+---+
6| | | | | | |PN| |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3|PB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Déplacement du PN de g7 vers g6

Test avec un déplacement de 2 cases :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
d4
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN| |PN|
+---+---+---+---+---+---+
6| | | | | | |PN| |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | |PB| | | | |
+---+---+---+---+---+---+
3|PB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB| |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Déplacement du PB de d2 vers d4

```
Veuillez choisir une case cible pour PN :
e5
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN| |PN| |PN|
+---+---+---+---+---+---+
6| | | | | | |PN| |
+---+---+---+---+---+---+
5| | | |PN| | | |
+---+---+---+---+---+---+
4| | |PB| | | | |
+---+---+---+---+---+---+
3|PB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB| |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Déplacement du PN de e7 vers e5

Maintenant si j'essaie de déplacer le pion PB situé en a3 de 2 cases, le déplacement est illégal et un message d'erreur devrait apparaître :

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|  |PN|  |PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |PN|  |
+---+---+---+---+---+---+
5|  |  |  |  |PN|  |  |  |
+---+---+---+---+---+---+
4|  |  |PB|  |  |  |  |  |
+---+---+---+---+---+---+
3|PB|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a3: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
a5
Coup illegal : Ce pion ne peut pas se déplacer comme ça
Veuillez choisir une case cible pour PB :

```

Il y a bien un message d'erreur qui annule le déplacement donc le système fonctionne. Maintenant, regardons ce qui se passe quand une pièce adverse se trouve devant le pion. Je vais faire des tests avec cette configuration :

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |PN|  |  |  |  |
+---+---+---+---+---+---+
4|  |  |PB|  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Regardons ce qui se passe si j'essaie d'avancer le pion blanc d'une case :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |PN|  |  |  |
+---+---+---+---+---+---+
4|  |  |  |PB|  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
d4: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
d5
Coup illegal : La case contient une piece adverse
Veuillez choisir une case cible pour PB :

```

Comme prévu le déplacement est refusé car la case cible contient une pièce noire. Maintenant, il faut vérifier que dans cette configuration, le pion peut se déplacer en c5 et en e5 :

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
c5
      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |PB|PN|  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
e5
      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |PN|PB|  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

C'est tout bon, les coups légaux permettent aux pions de se déplacer correctement. En revanche, il faut vérifier qu'un message d'erreur s'affiche lorsque le joueur souhaite déplacer le pion de manière illégale indépendamment du déplacement de 2 cases :

Dans ce test, j'essaie de déplacer le pion PB situé en d2 sur la case c5 qui est un coup illégal pour un pion.

```
C'est au tour des blancs. Quel est votre choix ?
d2

  a b c d e f g h
+---+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+  d2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
c5
Coup illegal : Ce pion ne peut pas se deplacer comme ça
Veuillez choisir une case cible pour PB :
█
```

Cela fonctionne, un message d'erreur s'affiche informant le joueur que le pion ne peut pas se déplacer de cette manière. Le joueur doit donc entrer une nouvelle case cible.

A présent le fonction MovePion est opérationnelle (du moins en apparence car en réalité il y a une faille dans le système de déplacement que j'explique en fin de rapport dans la section Amélioration et correction des bugs).

VI/ Déplacement de la tour (T)

Pour que le déplacement de la tour soit légal, elle doit bouger d'un nombre quelconque de cases verticalement ou horizontalement.

	a	b	c	d	e	f	g	h
8	1	2	3	4	5	6	7	8
7	9	10	11	12	13	14	15	16
6	17	18	19	20	21	22	23	24
5	25	26	27	28	29	30	31	32
4	33	34	35	36	37	38	39	40
3	41	42	43	44	45	46	47	48
2	49	50	51	52	53	54	55	56
1	57	58	59	60	61	62	63	64

	a	b	c	d	e	f	g	h
8								
7								
6								
5								
4					TB			
3								
2								
1								

Colonne de départ = colonne d'arrivée = 1

Ligne de départ = ligne d'arrivée = 8

Pour savoir le joueur tente de déplacer la tour à la verticale ou à l'horizontale, je vais créer 4 variables locales qui sont : *ld*, *la*, *cd*, *ca* correspondant respectivement aux valeurs de : la ligne de départ, la ligne d'arrivée, la colonne de départ et la colonne d'arrivée déterminées avec la variable *Coordonnees[8][8]*.

```
for(i = 0 ; i < 8 ; i = i+1)
{
    for(j = 0 ; j < 8 ; j = j+1)
    {
        if(Coordonnees[i][j] == depart)
        {
            ld = i+1;
            cd = j+1;
        }
        if(Coordonnees[i][j] == arrive)
        {
            la = i+1;
            ca = j+1;
        }
    }
}
```

Le coup sera légal si et seulement si $la = ld$ ou $cd = ca$ c'est à dire que les lignes de départ et d'arrivée sont les mêmes ou que les colonnes de départ et d'arrivé sont les mêmes et que la tour ne passe pas au dessus d'autres pièces. Pour savoir si la tour ne passe pas au dessus d'autre pièce, il faut que les sommets des piles se trouvant entre le départ et l'arrivée soit nul.

Voici un test pour un déplacement vertical :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TB :
a3
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | |PN|
+---+---+---+---+---+---+
4|PB| | | | | | |
+---+---+---+---+---+---+
3|TB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des noirs. Quel est votre choix ?

```

Déplacement de la TB de a1 vers a3

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TN :
h6
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN| |
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN| |
+---+---+---+---+---+---+
6| | | | | | |TN|
+---+---+---+---+---+---+
5| | | | | | |PN|
+---+---+---+---+---+---+
4|PB| | | | | | |
+---+---+---+---+---+---+
3|TB| | | | | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?

```

Déplacement due la TN de h8 vers h6

Test avec un déplacement horizontal :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TB :
f3
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN| |
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN| |
+---+---+---+---+---+---+
6| | | | | | |TN|
+---+---+---+---+---+---+
5| | | | | | |PN|
+---+---+---+---+---+---+
4|PB| | | | | | |
+---+---+---+---+---+---+
3| | | | |TB| | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des noirs. Quel est votre choix ?

```

Déplacement de la TB de d3 vers f3

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TN :
c6
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN| |
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN| |
+---+---+---+---+---+---+
6| |TN| | | | | |
+---+---+---+---+---+---+
5| | | | | | |PN|
+---+---+---+---+---+---+
4|PB| | | | | | |
+---+---+---+---+---+---+
3| | | | |TB| | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?

```

Déplacement de la TN de h6 vers c6

Les déplacements légaux pour la tour sont donc fonctionnels.

Maintenant regardons ce qui se passe quand on essaie de déplacer la tour en passant au-dessus d'autres pièces :

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4| | | | | | | |
+---+---+---+---+---+---+
3| | | | | | | |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a1: TB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TB :
a3
Coup illegal : vous ne pouvez pas passer au dessus d'une piece
Veuillez choisir une case cible pour TB :

```

déplacement de la tour de a1 vers a3

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN| |PN|PN|PN|
+---+---+---+---+---+---+
6| | | | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4|PB| | | | | | | |
+---+---+---+---+---+---+
3| |TB| | |PN| | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
b3: TB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TB :
e3
Coup illegal : la case contient une piece adverse
Veuillez choisir une case cible pour TB :
f3
Coup illegal : vous ne pouvez pas passer au dessus d'une piece
Veuillez entrer une nouvelle case cible :
Veuillez choisir une case cible pour TB :

```

déplacement de la tour de b3 vers e3 puis vers f3

Les déplacements illégaux sont donc évités et un message d'erreur est bien envoyé indiquant au joueur le problème et lui demande de choisir une autre case cible. Maintenant comme pour le pion, un message d'erreur doit s'afficher si le joueur tente de déplacer la tour autrement qu'à la verticale ou qu'à l'horizontale :

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7| |PN| |PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN| |PN| | | | | |
+---+---+---+---+---+---+
5| | | | | | | |
+---+---+---+---+---+---+
4|PB| | | | | | | |
+---+---+---+---+---+---+
3|TB| | | | | | | |
+---+---+---+---+---+---+
2| |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a3: TB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour TB :
e5
Coup illegal : La tour ne peut pas se deplacer de cette maniere
Veuillez choisir une case cible pour TB :

```

Un message d'erreur informe bien le joueur que la tour ne peut pas se déplacer comme cela et on lui demande de rentrer une nouvelle case cible.

Désormais la fonction MoveTour () est fonctionnelle, je vais donc passer au mouvement du fou.

VII/ Déplacement du fou (F)

Pour que le déplacement du fou soit légal, il doit bouger d'un nombre quelconque de cases en diagonale sans passer au dessus d'autres pièces.

	a	b	c	d	e	f	g	h
8	1	2	3	4	5	6	7	8
7	9	10	11	12	13	14	15	16
6	17	18	19	20	21	22	23	24
5	25	26	27	28	29	30	31	32
4	33	34	35	36	37	38	39	40
3	41	42	43	44	45	46	47	48
2	49	50	51	52	53	54	55	56
1	57	58	59	60	61	62	63	64

	a	b	c	d	e	f	g	h
8								
7								
6								
5								
4		FB						
3								
2								
1								

Pour pouvoir déplacer le fou, il me fallait trouver un moyen de savoir si le joueur tente de déplacer le fou sur une diagonale montante (flèche noire) ou descendante (flèche rouge). Après quelques raisonnements mathématiques, je me suis aperçu que :

- dans le cas d'un déplacement en diagonale descendante, la différence prise en valeur absolue de la valeur de la case d'arrivée et de départ était un multiple de 9.

Exemple : pour la flèche rouge, arrive – départ = 52 – 34 = 18 = 2*9

Implémentation en C : `if((int)fabs(arrive - depart) % 9 == 0)`

- dans le cas d'un déplacement en diagonale montante, la différence prise en valeur absolue de la valeur de la case d'arrivée et de départ était un multiple de 7.

Exemple : pour la flèche noire, arrive – départ = |20 – 34| = 14 = 2*7

Implémentation en C : `if((int)fabs(arrive - depart) % 7 == 0)`

Avec ces deux conditions j'ai réussi à savoir sur quelle diagonale le joueur veut déplacer le fou.

Pour savoir si le fou ne passe pas au-dessus d'une pièce lors de son déplacement, j'ai utilisé la même méthode que pour le déplacement de la tour.

Voici un test pour un déplacement en diagonale montante :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FB :
f4
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |PB|  |FB|  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|  |DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Déplacement de la FB de c1 vers f4

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FN :
c5
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|  |CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|  |PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |FN|  |PN|  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|PB|PB|  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Déplacement due la FN de f8 vers c5

Test avec un déplacement en diagonale descendante :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FB :
a3
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|PB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|FB|  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|  |DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Déplacement de la FB de c1 vers a3

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FN :
g4
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|  |DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |PN|  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |FN|  |  |
+---+---+---+---+---+---+
3|PB|PB|  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Déplacement due la FN de c8 vers g4

Les déplacements légaux pour le fou sont donc fonctionnels.

Maintenant regardons ce qui se passe quand on essaie de déplacer le fou en passant au-dessus d'autres pièces ou que la case ciblée contient une pièce adverse :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|  |PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |PN|  |
+---+---+---+---+---+---+
4|  |  |PB|  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+  c1: FB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FB :
g5
Coup illegal : la case contient une piece adverse
Veuillez choisir une case cible pour FB :
h6
Coup illegal : vous ne pouvez pas passer au dessus d'une piece
Veuillez choisir une case cible pour FB :

```

Je cherche à déplacer le FB en c1 vers g5 puis après vers h6

Les déplacements illégaux sont donc évités et un message d'erreur est bien envoyé indiquant au joueur le problème et lui demandant de choisir une autre case cible. Maintenant comme pour toutes les pièces du jeu, un message d'erreur doit s'afficher si le joueur tente de déplacer le fou autrement qu'en diagonale :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|  |  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|PN|PN|PN|  |  |  |  |
+---+---+---+---+---+---+
4|PB|PB|PB|  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|  |  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+  c1: FB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour FB :
c3
Coup illegal : Le fou ne peut pas se deplacer de cette maniere
Veuillez choisir une case cible pour FB :

```

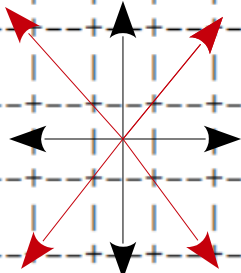
Un message d'erreur informe bien le joueur que la tour ne peut pas se déplacer comme ça et on lui demande de rentrer une nouvelle case cible.

Désormais la fonction MoveFou () est fonctionnelle, je vais donc passer au mouvement de la dame et du roi.

VIII/ Déplacement de la dame (D) et roi (R)

Pour que le déplacement de la dame soit légal, elle doit bouger d'un nombre quelconque de cases verticalement, horizontalement ou en diagonale. Le roi, quant à lui, bouge de la même manière que la dame mais son déplacement est limité à une seule case.

	a	b	c	d	e	f	g	h		a	b	c	d	e	f	g	h	
	+--+--+--+--+--+--+--+--+									+--+--+--+--+--+--+--+--+								
8	1	2	3	4	5	6	7	8		8	TN	CN	FN	DN	RN	FN	CN	TN
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
7	9	10	11	12	13	14	15	16		7	PN	PN	PN	PN	PN	PN	PN	PN
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
6	17	18	19	20	21	22	23	24		6								
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
5	25	26	27	28	29	30	31	32		5								
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
4	33	34	35	36	37	38	39	40		4								
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
3	41	42	43	44	45	46	47	48		3								
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
2	49	50	51	52	53	54	55	56		2	PB	PB	PB	PB	PB	PB	PB	PB
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							
1	57	58	59	60	61	62	63	64		1	TB	CB	FB	DB	RB	FB	CB	TB
	+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+							



Après réflexion, il est devenu évident que le déplacement de la dame et du roi rassemblaient les déplacements de la tour (flèches noires) et du fou (flèches rouges) réalisés un peu plus tôt. Donc si le joueur tente de déplacer la dame comme une tour, alors j'appelle la fonction MoveTour () et si le joueur tente de déplacer la dame comme un fou, alors j'appelle la fonction MoveFou ().

Implémentation en C :

```

if((fabs (arrive - depart) < 8 && la == ld) || (fabs (arrive - depart) >= 8 && ca == cd))
{
    MoveTour();
}

else if((int)fabs(arrive - depart) % 9 == 0 || (int)fabs(arrive - depart) % 7 == 0)
{
    MoveFou();
}

if(strchr(sommet(currentPile), 'D') != NULL)
{
    printf("Coup illegal : La Dame ne peut pas se deplacer comme ça\n");
    MovePiece();
}
else if(strchr(sommet(currentPile), 'R') != NULL)
{
    printf("Coup illegal : Le Roi ne peut pas se deplacer comme ça\n");
    MovePiece();
}

```

Voici un test pour un déplacement comme une tour :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour DB :
d3
  a b c d e f g h
+-----+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+-----+
7|  |PN|PN|PN|PN|PN|PN|PN|
+-----+
6|PN|  |  |  |  |  |  |
+-----+
5|  |  |  |  |  |  |  |
+-----+
4|  |  |PB|  |  |  |  |
+-----+
3|  |  |DB|  |  |  |  |
+-----+
2|PB|PB|PB|  |PB|PB|PB|PB|
+-----+
1|TB|CB|FB|  |RB|FB|CB|TB|
+-----+
C'est au tour des noirs. Quel est votre choix ?
```

Déplacement de la DB de d1 vers d3

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour RB :
e2
  a b c d e f g h
+-----+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+-----+
7|  |PN|PN|PN|PN|PN|PN|PN|
+-----+
6|PN|  |  |  |  |  |  |
+-----+
5|  |  |  |  |  |  |  |
+-----+
4|  |  |  |PB|  |  |  |
+-----+
3|  |  |  |  |  |  |  |
+-----+
2|PB|PB|PB|PB|RB|PB|PB|PB|
+-----+
1|TB|CB|FB|DB|  |FB|CB|TB|
+-----+
C'est au tour des noirs. Quel est votre choix ?
```

Déplacement du RB de e1 vers e2

Test avec un déplacement comme un fou :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour DB :
h5
  a b c d e f g h
+-----+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+-----+
7|  |PN|PN|PN|PN|PN|PN|PN|
+-----+
6|PN|  |  |  |  |  |  |
+-----+
5|  |  |  |  |  |  |DB|
+-----+
4|  |  |  |PB|  |  |  |
+-----+
3|  |  |  |  |  |  |  |
+-----+
2|PB|PB|PB|PB|  |PB|PB|PB|PB|
+-----+
1|TB|CB|FB|  |RB|FB|CB|TB|
+-----+
C'est au tour des noirs. Quel est votre choix ?
```

Déplacement de la DB de d1 vers h5

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour RB :
f2
  a b c d e f g h
+-----+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+-----+
7|  |PN|PN|PN|PN|PN|PN|PN|
+-----+
6|  |PN|  |  |  |  |  |
+-----+
5|PN|  |  |  |  |  |  |
+-----+
4|  |  |  |  |PB|  |  |
+-----+
3|  |  |  |  |  |  |  |
+-----+
2|PB|PB|PB|PB|PB|RB|PB|PB|PB|
+-----+
1|TB|CB|FB|DB|  |FB|CB|TB|
+-----+
C'est au tour des noirs. Quel est votre choix ?
```

Déplacement due la RB de e1 vers f2

Les déplacements légaux pour la dame et le roi sont donc fonctionnels.

Maintenant regardons ce qui se passe quand on essaie de déplacer la dame ou le roi en passant au-dessus d'autres pièces ou que la case ciblée contient une pièce adverse :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|  |  |
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |PB|  |PN|PN|
+---+---+---+---+---+---+
3|PB|  |PB|  |PB|  |  |
+---+---+---+---+---+---+
2|  |PB|  |  |PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+  d1: DB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour DB :
g4
Coup illegal : la case contient une piece adverse
Veuillez choisir une case cible pour DB :
h5
Coup illegal : vous ne pouvez pas passer au dessus d'une piece
Veuillez choisir une case cible pour DB :

```

Je cherche à déplacer le DB en d1 vers g4 puis après vers h5

Les déplacements illégaux sont donc évités et un message d'erreur est bien envoyé indiquant au joueur le problème et lui demandant de choisir une autre case cible. Maintenant comme pour toutes les pièces du jeu, un message d'erreur doit s'afficher si le joueur tente de déplacer la dame autrement qu'en diagonale ou à la verticale ou à l'horizontale :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |PB|  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|  |PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+  d1: DB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour DB :
e3
Coup illegal : La Dame ne peut pas se deplacer comme ça
Veuillez choisir une case cible pour DB :

```

Un message d'erreur informe bien le joueur que la tour ne peut pas se déplacer comme ça et on lui demande de rentrer une nouvelle case cible.

Un message d'erreur doit également apparaître si on tente de déplacer un roi de plus de 2 cases :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |PB|PB|  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|  |PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+  e1: RB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour RB :
e3
Coup illegal : Le Roi ne peut pas se deplacer de cette maniere
Veuillez choisir une case cible pour RB :
g3
Coup illegal : Le Roi ne peut pas se deplacer de cette maniere
Veuillez choisir une case cible pour RB :
```

Un message d'erreur informe bien le joueur que le roi ne peut pas se déplacer comme ça et on lui demande de rentrer une nouvelle case cible. Conclusion la fonction MoveDameEtRoi () fonctionne parfaitement. Il ne reste plus qu'a implémenter la fonction de déplacement du cavalier.

IX/ Déplacement du cavalier (C)

Pour que le déplacement du cavalier soit légal, il doit se déplacer en L c'est à dire une case verticalement et deux cases horizontalement, ou deux cases verticalement et une case horizontalement.

	a	b	c	d	e	f	g	h		a	b	c	d	e	f	g	h	
8										8								
7										7								
6				-17		-15				6			-17		-15			
5			-10				-6			5			-10				-6	
4					CB					4					CB			
3			+6				+10			3			+6				+10	
2				+15		+17				2				+15		+17		
1										1								

Pour trouver une relation entre les valeurs des cases de l'échiquier et des cases possibles de déplacement d'un cavalier, j'ai placé un cavalier au centre du plateau de jeu ici en e4. Donc la valeur de départ est égale à 37. Puis pour chaque case où le déplacement du cavalier est légal, j'y ai inscrit la valeur de la différence entre la valeur de cette case et celle de départ. Il est apparu que le cavalier peut se déplacer sur les cases où la différence entre les variables *arrive* et *départ* prit en valeur absolue est égale à 6 ou 10 ou 15 ou 17. De plus, à la différence des autres pièces du jeu, le cavalier a la possibilité de passer au-dessus d'autres pièces mais ne peut cependant pas se déplacer sur une case contenant une pièce adverse.

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour CB :
c3
  a b c d e f g h
+---+---+---+---+---+---+
8| TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7| PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  | CB|  |  |  |  |  |
+---+---+---+---+---+---+
2| PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| TB|  |FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des noirs. Quel est votre choix ?

```

Déplacement du CB de b1 vers c3

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour CN :
f6
  a b c d e f g h
+---+---+---+---+---+---+
8| TN|CN|FN|DN|RN|FN|  |TN|
+---+---+---+---+---+---+
7| PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  | CN|  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  | CB|  |  |  |  |  |
+---+---+---+---+---+---+
2| PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1| TB|  |FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?

```

Déplacement du CN de g8 vers f6

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour CB :
e4
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|  |TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |CN|  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |CB|  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|  |FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Déplacement du CB de c3 vers e4

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour CN :
d5
  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|  |TN|
+---+---+---+---+---+---+
7|PN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |CN|  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |CB|  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|  |FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Déplacement du CN de f6 vers d5

Les déplacements légaux pour le cavalier sont donc fonctionnels. Il faut maintenant vérifier qu'un message d'erreur apparaît quand on essaie de déplacer le cavalier sur une pièce adverse ou que l'on essaie de déplacer le cavalier de manière illégale :

```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PN|PN|PN|  |PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |PN|  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |CB|  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|  |FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
c3: CB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour CB :
d5
Coup illegal : la case contient une piece adverse
Veuillez choisir une case cible pour CB :
d4
Coup illegal : Le Cavalier ne peut pas se deplacer comme ça
Veuillez choisir une case cible pour CB :

```

Lorsqu' on essaie de déplacer le cavalier en partant de c3 vers d5 ou vers d4, des messages d'erreurs informent le joueur que le cavalier ne peut pas se déplacer comme ça ou qu'une pièce adverse se trouve sur la case cible et on lui demande de rentrer une nouvelle case cible.

Maintenant toutes les fonctions pour déplacer les pièces sont fonctionnelles. Il faut maintenant gérer l'empilement et le dépilement des piles.

X/ Utilisation et mise en place des piles

A chaque fois que le déplacement d'une pièce est possible, il faut désempiler la pièce de la pile de départ afin de l'empiler au sommet de la pile cible. C'est le rôle de la fonction que j'ai créée qui s'appelle `PopAndPush(int i)` qui attend en paramètre un entier `i` :

- Si `i` est égal à 0, on indique à la fonction que l'on souhaite désempiler la pile de départ.
- Si `i` est différent de 0, on indique à la fonction que l'on souhaite empiler le sommet de la pile de départ à la pile d'arrivée.

Implémentation en C :

```
while(strcmp(&coup,Echiquier[j]) != 0)
{
    j=j+1;
}
pop(&T[j]);
s = sommet(T[j]);
if(!is_empty(T[j]))
    Piece[depart - 1] = s;
```

} Identifie et supprime le sommet `s` de la pile de départ

} Si après suppression de son sommet, la pile de départ n'est pas vide alors le nouveau sommet sera affiché lors de l'actualisation de l'affichage

```
while(strcmp(&cible,Echiquier[j]) != 0)
{
    j=j+1;
}
```

} Identifie la pile d'arrivée

```
e = sommet(currentPile);
push(e,&T[j]);
```

} Empile le sommet de la pile de départ sur la pile d'arrivée

Maintenant que le module pour empiler les pièces est mis en place, il faut maintenant tester si deux pièces d'une même couleur peuvent s'empiler et que l'affilage de la pile est fonctionnel. Voici mon premier test :

```
  a b c d e f g h
+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+

La case choisie ne contient pas de piece
C'est au tour des blancs. Quel est votre choix ?
```

Ici j'ai empilé la TB en a1 sur le pion PB présent en a2.

Quand je veux sélectionner la case a1, un message me dit que la case est vide.

Cela veut donc dire que la pile a1 ne contient plus de pièce et a donc bien été désempilée.


```

  a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|      TB
+---+---+---+---+---+---+      a2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :

```

Maintenant si je sélectionne la case a2, je vois que celle-ci contient bien 2 pièces qui sont TB et PB

L'empilage à donc bien fonctionné

Dans ce cas j'ai la possibilité de choisir un nombre de pièces à déplacer

Maintenant que la fonction `PopAndPush` fonctionne, il faut s'assurer que le joueur puisse entrer un nombre de pièces à déplacer et de ce nombre est d'une part différent de 0 et d'autre part inférieur ou égal à la taille de la pile de départ.

Pour cela, le nombre de pièces que le joueur aura choisi sera stocké dans la variable *PieceDeplace*. Néanmoins, il faut pouvoir s'assurer que le joueur rentre bien un entier.

J'ai donc créé une variable de type `char` *NombreDePiece* qui va récupérer le nombre entré par le joueur par intermédiaire de la fonction `scanf` puis la valeur de *PieceDeplace* sera le résultat de la fonction `atoi` de la librairie `stdlib` qui convertit un caractère `char` en un entier `int` :

```

printf("Veuillez choisir un nombre de piece a deplacer :\n");
scanf("%s",&NombreDePiece);
PieceDeplace = atoi(&NombreDePiece);

```

Il faut ensuite s'assurer que *PieceDeplace* soit un nombre différent de 0 et inférieur ou égal à la taille de la pile de départ :

```

if(PieceDeplace == 0)
{
    printf("Nombre de piece incorrects :\n");
    Test();
}

if(PieceDeplace >= l+1)
{
    printf("La case contient %i vous ne pouvez donc pas déplacer %i pièces\n",l, PieceDeplace);
    Test();
}

```

Si ces conditions ne sont pas respectées, on rappelle la fonction `Test` pour demander au joueur de choisir un nouveau nombre de pièces à déplacer.

Je vais à présent tester ces deux conditions :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|      TB
+---+---+---+---+---+---+      a2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir un nombre de piece a deplacer :
0
Nombre de piece incorrects :
Veuillez choisir un nombre de piece a deplacer :
3
La case contient 2 vous ne pouvez donc pas déplacer 3 pièces
Veuillez choisir un nombre de piece a deplacer :
```

La pile a2 contient 2 pièces. Dans un premier temps, je décide de ne déplacer aucune pièce et le message d'erreur apparaît et on me propose de choisir un nouveau nombre de pièces à déplacer. Je décide cette fois de déplacer 3 pièces alors que la pile n'en contient que 2. Un message d'erreur apparaît et me demande une nouvelle fois de choisir un nouveau nombre de pièces à déplacer.

Si le joueur décide de déplacer plusieurs pièces, il faut que je crée une variable *CoupRestant* pour savoir combien de pièces il lui reste à déplacer et qui sera décrémenté de 1 à chaque fois qu'une pièce sera déplacée. Donc initialement *CoupRestant* sera égale à *PieceDeplace* et tant que *CoupRestant* sera différent de 0, le joueur peut continuer à déplacer des pièces.

Je vais faire un test avec l'exemple ci-dessus : je vais choisir de déplacer 2 pièces qui sont la TB et le PB. Dans un premier temps, je vais déplacer la tour TB sur la case a4 puis le pion PB sur la case a3 et voir le résultat obtenu :

Sélection du nombre de pièces à déplacer :

```

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir un nombre de piece a deplacer :
2
Veuillez choisir une case cible pour TB :
```

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|TB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|PB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a2: PB

Veillez choisir une case cible pour PB :

```

La tour TB s'est bien déplacée et la pile a2 s'est désempilée. Il ne reste donc plus que le pion PB dans la pile a2. Puisqu' on me demande bien de choisir une case pour déplacer ce pion, le système et l'affichage fonctionnent donc correctement.

```

Veillez choisir une case cible pour PB :
a3
      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
4|TB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|PB|  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Le pion c'est lui aussi déplacé correctement et une fois que j'ai déplacé le nombre de pièce que je souhaitait, on change bien de joueur l'affichage se rafraîchir correctement.

A ce stade du projet il ne manque plus qu'a implanter un système pour pouvoir prendre des pièces de l'adversaire car il ne suffit pas de se déplacer sur une case contenant une pièce adverse pour la prendre ; il faut déplacer un nombre de pièces strictement supérieur à celui des pièces adverses dans la case atteinte.

Cela veut dire qu'il est possible de se rendre sur une case contenant une pièce ou des pièces adverses si et seulement si la taille de la pile cible est strictement inférieure à la valeur de *PieceDeplace*.

Si c'est le cas, il faut vider la pile cible puis lui empiler le sommet de la pile de départ. Pour l'implémenter, j'ai rajouté ces lignes de code dans la fonction PopAndPush :

```
if(PieceDeplace > TaillePile(T[j]) && ((Joueur == 1 && strchr(sommet(T[j]), 'N') != NULL) || (Joueur == 0 && strchr(sommet(T[j]), 'B') != NULL)))
{
    while(!is_empty(T[j]))
    {
        pop(&T[j]);
        if(Joueur == 1)
            PieceNrestantes = PieceNrestantes-1;
        else
            PieceBrestantes = PieceBrestantes-1;
    }
}
```

En revanche, si un joueur tente de prendre des pièces de l'adversaire alors qu'il ne déplace pas plus de pièce qu'il n'y a de pièce de l'adversaire sur la case cible , il faut envoyer un message au joueur afin qu'il choisisse une nouvelle case cible.

Je vais maintenant tester mon système en partant de l'état ci-dessous :

```

      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8| TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+---+
5|  |  |  |  |  |  |  |
+---+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|  |CB|FB|DB|RB|FB|CB|TB|      TB
+---+---+---+---+---+---+---+      a2: PB

Que souhaitez vous faire ? veuillez entrer une lettre :

```

Je devrais être en mesure de prendre le pion PN situé en a6 en déplaçant les 2 pièces TB et PB situées en a2 et de m'assurer une fois cette opération effectuée que la case a6 ne contient plus le pion PN.

```

      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8 | TN | CN | FN | DN | RN | FN | CN | TN |
+---+---+---+---+---+---+---+
7 |   | PN | PN | PN | PN | PN | PN | PN |
+---+---+---+---+---+---+---+
6 | TB |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3 | PB |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2 |   | PB | PB | PB | PB | PB | PB | PB |
+---+---+---+---+---+---+---+
1 |   | CB | FB | DB | RB | FB | CB | TB |
+---+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Voilà l'affichage à la fin de l'opération et la tour TB a bien remplacé le pion PN. Cependant il faut s'assurer qu'il n'y a pas eu de problème et que le pion PN a bien été désempilée de la pile a6 :

```

C'est au tour des noirs. Quel est votre choix ?
a6
      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+
8 | TN | CN | FN | DN | RN | FN | CN | TN |
+---+---+---+---+---+---+---+
7 |   | PN | PN | PN | PN | PN | PN | PN |
+---+---+---+---+---+---+---+
6 | TB |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4 |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3 | PB |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2 |   | PB | PB | PB | PB | PB | PB | PB |
+---+---+---+---+---+---+---+
1 |   | CB | FB | DB | RB | FB | CB | TB |
+---+---+---+---+---+---+---+
a6: TB

```

C'est tout bon, maintenant il est possible de prendre des pièces de l'adversaire lorsque le nombre de pièces déplacées est strictement supérieur au nombre de pièces présentes sur la pile cible.

Il faut maintenant s'assurer qu'il n'est pas possible de prendre des pièces de l'adversaire lorsque le nombre de pièces déplacées est inférieur ou égale au nombre de pièces présent sur la pile cible. Pour cela, voici mon test :

```

      a b c d e f g h
+---+---+---+---+---+---+---+
8|   |CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|TN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|   |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Cette fois ci les case a2 et a7 contiennent 2 pièces respectivement TB et PB puis TN et PN. Il n'est donc théoriquement pas possible de déplacer la tour TB en a2 sur la case a7. Regardons le résultat de cette opération :

```

      a b c d e f g h
+---+---+---+---+---+---+---+
8|   |CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+---+
7|TN|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+---+
6|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
5|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
4|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
3|   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+---+
1|   |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+---+

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir un nombre de piece a deplacer :
2
Veuillez choisir une case cible pour TB :
a7
Coup illegal : la case contient une piece adverse
Veuillez choisir une case cible pour TB :

```

Il est vrai que le message d'erreur n'est pas très parlant mais il indique qu'il n'est pas possible de déplacer des pièces vers la case a7 donc le système est fonctionnel.

XI/ Amélioration et correction des bugs

Fin de la partie :

Pour savoir à quel moment la partie se termine, j'ai créé 2 variables de type int : *PieceBrestantes* et *PieceNrestantes* qui sont initialement toutes les deux égales à 16 car il y a au début 16 pièces noires et 16 pièces blanches sur le plateau de jeu.

Ces deux variables sont mises à jour dans la fonction *PopAndPush ()*. A chaque fois qu'un joueur prend des pièces de l'adversaire, le nombre de pièces de l'adversaire diminue à chaque fois que l'on retire une pièce de la pile cible jusqu'à ce que celle-ci soit vide.

```
while(!is_empty(T[j]))
{
    pop(&T[j]);
    if(Joueur == 1)
        PieceNrestantes = PieceNrestantes-1;
    else
        PieceBrestantes = PieceBrestantes-1;
}
```

A chaque fois que l'on change de joueur, je vérifie que chaque joueur possède au moins une pièce sur le plateau de jeu. Si un joueur perd toutes ses pièces alors la partie se termine et informe l'autre joueur qu'il vient de remporter la partie.

```
if(PieceBrestantes != 0 && PieceNrestantes != 0)

else if(PieceBrestantes == 0)
{
    printf("Les noirs ont gagnés la partie\n");
    exit(1);
}
else if(PieceNrestantes == 0)
{
    printf("Les blancs ont gagnés la partie\n");
    exit(1);
}
```

Transformation d'un pion en dame:

Pour qu'un pion se transforme en dame, il doit arriver sur la dernière rangée (rangée n° 8 pour les pions blancs et rangée n°1 pour les pions noirs).

Pour savoir si un pion vérifie cette condition, j'ai décidé de créer une variable de type int k. k est égale à 1 si le pion arrive sur la dernière rangée sinon elle est égale à 0.

A chaque fois qu'un joueur décide de bouger un pion, je regarde si le pion a atteint la dernière rangée à l'aide d'une boucle for et de la variable *Coordonnées[8][8]* . Si je détecte qu'un pion noir à atteint la rangée n°1 :

```
k=0;
for(i = 0; i < 8; i=i+1)
{
    if(Coordonnees[7][i] == arrive)
        k = 1;
}
```

ou qu'un pion blanc a atteint la rangée n°8 :

```
k=0;
for(i = 0; i < 8; i=i+1)
{
    if(Coordonnees[0][i] == arrive)
        k = 1;
}
```

Dans ce cas, ma variable k passe a 1 et sera utilisée dans la fonction PopAndPush(). Pour cela, il faut retirer le pion de la pile cible puis lui ajouter une dame blanche ou noir selon le joueur. J'ai donc écrit une condition qui empile une pièce dame selon le joueur et la valeur de k :

```
if(Joueur == 1 && k != 0)
{
    PieceBrestantes = PieceBrestantes + 1;
    push("DB",&T[j]);
}
else if(Joueur == 0 && k != 0)
{
    PieceNrestantes = PieceNrestantes + 1;
    push("DN",&T[j]);
}
```


Voici mes tests pour m'assurer du bon fonctionnement du système : j'ai fait en sorte de libérer la place pour amener volontairement de pion PB en a7

```

      a b c d e f g h
+---+---+---+---+---+---+
8|   |CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|PB|PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
5|   |   |   |   |   |TN|   |
+---+---+---+---+---+---+
4|   |PN|   |   |   |   |   |
+---+---+---+---+---+---+
3|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|   |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

Je vais maintenant déplacer le pion PB de a7 vers a8 afin de voir le résultat :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|DB|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|   |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
5|   |   |   |   |   |TN|   |
+---+---+---+---+---+---+
4|   |PN|   |   |   |   |   |
+---+---+---+---+---+---+
3|   |   |   |   |   |   |   |
+---+---+---+---+---+---+
2|TB|PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|   |CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

a8: DB

Que souhaitez vous faire ? veuillez entrer une lettre :

```

La pièce PB en a8 s'est bien transformée en DB et une dame blanche a bien été empilée au sommet de la pile a8, donc le système est fonctionnel.

Explication du bug lors d'un déplacement du pion :

Comme je l'ai évoqué précédemment, il existe un bug dans le mouvement du pion que seul un joueur qui connaît la manière dont est codé un jeu d'échecs pourrait connaître cette faille. Pour l'illustrer je vais la mettre en pratique avec cette situation :

```
      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
4|PB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des blancs. Quel est votre choix ?

```

C'est au tour des blancs de jouer. Si je décide de déplacer le pion PB situé en a4 alors la seule solution possible est de le déplacer en b5 étant donné que la case a5 est occupée par une pièce noire. Néanmoins, regardez ce qui se passe si (intentionnellement ou par erreur) le joueur sélectionne la case h6 comme case cible :

```
Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
h6
      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |PB|
+---+---+---+---+---+---+
5|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
4|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+

C'est au tour des noirs. Quel est votre choix ?

```

Zut..... Le pion s'est bel et bien déplacé alors que ce type de déplacement est totalement illégal. En réalité d'un point de vue codage c'est totalement logique et voici pourquoi :

	a	b	c	d	e	f	g	h		a	b	c	d	e	f	g	h	
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
8	1	2	3	4	5	6	7	8		8	TN	CN	FN	DN	RN	FN	CN	TN
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
7	9	10	11	12	13	14	15	16		7		PN	PN	PN	PN	PN	PN	PN
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
6	17	18	19	20	21	22	23	24		6								
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
5	25	26	27	28	29	30	31	32		5	PN							
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
4	33	34	35	36	37	38	39	40		4	PB							
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
3	41	42	43	44	45	46	47	48		3								
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
2	49	50	51	52	53	54	55	56		2		PB	PB	PB	PB	PB	PB	PB
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+
1	57	58	59	60	61	62	63	64		1	TB	CB	FB	DB	RB	FB	CB	TB
	+	-	+	-	+	-	+	-	+	+	-	+	-	+	-	+	-	+

Dans le cas où la case visée contient au moins une pièce adverse le mouvement légal d'un pion est un mouvement d'une case en diagonale en direction de l'adversaire.

Le pion PB se trouve en a4 qui a pour numéro 33. donc si on ne résonne que sur le numéro des cases, les deux cases d'arrivée possible du pion seraient les cases 24(h6) et 26(b5).

Seulement dans ce cas présent, le pion PB se trouve sur le bord gauche, il n'existe donc pas de case à gauche du pion PB.

La case h6 numérotée 26 est donc illégale mais rien n'interdit le joueur de rentrer h6 comme case d'arrivée étant donnée qu'elle constitue un déplacement légal si le pion ne s'était pas trouvé sur le bord.

En cherchant à corriger ce problème, je me suis aperçu que ce déplacement illégal obligeait le pion à sauter par dessus une ligne ce qui n'est pas possible.

Pour y remédier, j'aurais créé deux variables pour stocker la ligne de départ et la ligne d'arrivée lorsqu'un joueur tente de déplacer un pion afin de m'assurer que la différence entre la ligne de départ et d'arrivée prise en valeur absolue est bien égale à 1.

Pour valider ma correction, j'ai réitéré exactement la même manipulation que celle effectuée ci-dessus pour mettre en évidence le problème :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
4|PB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
C'est au tour des blancs. Quel est votre choix ?

```

Et voici le résultat après avoir essayé de déplacer le pion PB de a4 vers h6 :

```

      a b c d e f g h
+---+---+---+---+---+---+
8|TN|CN|FN|DN|RN|FN|CN|TN|
+---+---+---+---+---+---+
7|  |PN|PN|PN|PN|PN|PN|PN|
+---+---+---+---+---+---+
6|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
5|PN|  |  |  |  |  |  |
+---+---+---+---+---+---+
4|PB|  |  |  |  |  |  |
+---+---+---+---+---+---+
3|  |  |  |  |  |  |  |
+---+---+---+---+---+---+
2|  |PB|PB|PB|PB|PB|PB|PB|
+---+---+---+---+---+---+
1|TB|CB|FB|DB|RB|FB|CB|TB|
+---+---+---+---+---+---+
a4: PB

Que souhaitez vous faire ? veuillez entrer une lettre :
d
Veuillez choisir une case cible pour PB :
h6
Coup illegal : Ce pion ne peut pas se deplacer comme ça
Veuillez choisir une case cible pour PB :

```

Cette fois le déplacement ne s'est pas effectué et un message d'erreur s'est bien affiché demandant au joueur de choisir une autre case cible. La correction apportée a donc permis de corriger le défaut de déplacement du pion.

XI/ Conclusion et limite du programme

J'ai déjà eu l'occasion de programmer de petits jeux en langage C# mais jamais en C, j'ai pris beaucoup de plaisir à réaliser SEUL ce jeu d'échecs modifié.

Cependant, avec davantage de temps j'aurais pu effectuer tout ce qui était attendu dans le cahier des charges.

Au début, je suis parti d'un plateau de jeu d'échecs classique ainsi qu'avec les mêmes dispositions de pièces pour pouvoir implanter le cavalier plus rapidement en me disant que je ferai la généralisation de taille $n \times n$ où l'on demande au joueur de choisir la valeur de n un peu plus tard.

Mais je n'ai pas pu finaliser cette généralisation alors j'ai préféré laisser la disposition initiale. En effet, je n'ai pas réussi à créer un tableau de taille $n \times n$ qui pourrait contenir toutes les piles du plateau de jeu même en utilisant des allocations dynamiques.

La deuxième chose est que dans l'extension proposée, mes recherches n'ont pas abouti au codage de la particularité du cavalier qui a la possibilité de porter une autre pièce. Je n'avais pas de solution pertinente pour implémenter cette particularité.

Malgré cela, le code du jeu compile sans erreur avec les extensions demandées et toutes les autres fonctions qui m'étaient demandées et elles sont toutes fonctionnelles. Cela m'a permis de réaliser plusieurs parties test contre des membres de ma famille afin de m'assurer du bon fonctionnement du jeu me permettant de repérer quelques erreurs par la suite corrigées sans problème.