

# PROGET 1

Jeux d'échec mobile

## RESUME

L'objectif est de créer un jeux d'échec mobile en ligne ou des joueurs s'affronterons en 1v1 les deux joueurs seront sur deux téléphones différents. Dans ce projet j'ai défini deux types d'objectifs, les objectifs principaux qui sont les choses qui seront forcément présente dans le projet et les objectifs secondaire qui eux seront ajoutés si j'ai le temps de les ajouter (car non obligatoires).

Frederic Macabiau

Objectifs

## Table des matières

1. Objectif principal .....	2
1.1. Interface Graphique : .....	2
1.2. Gameplay : .....	2
1.2.1. Déplacement des pieces : .....	3
1.2.2. Comment les pieces capturent les pièces des adversaires : .....	4
1.2.3. Règles spécifiques : .....	5
1.2.4. Fin de partie .....	5
1.2.5. Notation de la partie : .....	5
2. Objectifs secondaires : .....	7
2.1. Authentification : .....	7
2.2. Scorage : .....	7
2.3. Points d'XP : .....	7
2.4. Note de bienséance : .....	7
2.5. Mode spectateur : .....	7
2.6. Protection anti-tcheat : .....	7
3. Quelques approfondissements Techniques : .....	8
3.1. Multi-Threading : .....	8
3.1.1. Lancement de L'application : .....	8
3.1.2. Creation d'une partie : .....	8
3.1.3. Lorsque vous jouez un coups : .....	9
3.2. Programmation Orientée Objets : .....	10
3.3. Interactions entre L'ihm, le serveur et la classe Pièce .....	12

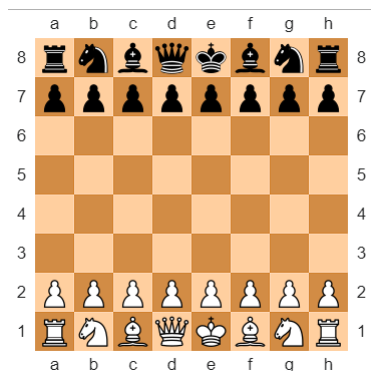
## 1. Objectif principal

L'objectif principal sera simplement de créer un jeu d'échec simple.

### 1.1. Interface Graphique :

L'interface du menu contient une liste des parties disponible. Avec un bouton jouer pour rejoindre la partie. Ainsi qu'un bouton pour créer une nouvelle partie.

L'interface au sein d'une partie pour chaque joueurs contiendra un échiquier vu d'au dessus (en 2D) sur lequel il y aura nos pièces et celles de notre adversaire (en 2D), l'image des pièces sera à l'intérieur des cases de l'échiquier sur lesquelles ils sont.



Le placement ici présente représente le placement des pièces au début de la partie (avec un joueur ayant les pions blanc attribués).

Au dessus de l'échiquier il y aura quatre zone de texte. Deux au centre, dont une indiquant quel joueur doit jouer et l'autre indiquant la valeur d'arbitrage\*. Les deux autres seront à droite dont une indiquant le temps restant pour jouer votre coup, et l'autre indiquant la durée de la partie.

Et enfin il y aura un bouton en haut à gauche pour quitter la partie.

### 1.2. Gameplay :

Lorsque vous arrivez dans le jeu vous avez un menu, avec une liste des parties que vous pouvez rejoindre, en tant que joueur si il n'y a qu'un seul joueur (partie en attente).

Ainsi les parties attendant un second joueur sont en attentes et les parties ayant deux joueurs (sont donc en cours).

Vous pouvez aussi créer votre propre partie et attendre qu'un autre joueur vous rejoigne.

Des que vous arrivez sur la partie la couleur des pièces (blanc ou noir) vous sont attribuée de manière aléatoire. Et la partie se lance dès qu'il y a deux joueurs.

Lorsque c'est à votre tour de jouer un message en haut de l'écran s'affichera pour vous l'indiquer « c'est à votre tour ». Vous pourriez sélectionner la pièce que vous souhaitez jouer en cliquant sur la case dans laquelle elle se situe. Une fois celle-ci sélectionnée, les cases sur lesquelles elle pourra se déplacer seront affichées en vert et les cases contenant des pièces que vous pouvez capturer seront affichées en rouge. A partir de ce moment vous pourriez soit cliquer sur une de ces cases pour déplacer votre pièce, ou bien cliquer sur une autre pièce pour changer de pièce.

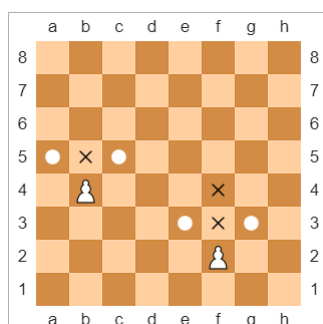
Chaque joueur aura un délai 20 minutes maximum pour jouer un coup (temps que j'ai estimé largement suffisant pour réfléchir au coup que l'on va jouer, tout sans trop ralentir la partie).

## PROGET 1

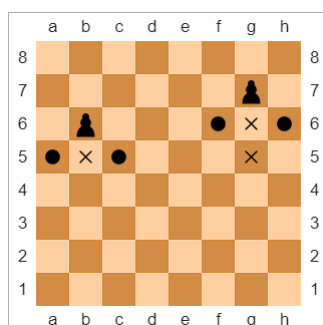
Les règles du jeu sont bien entendu celles des échecs ainsi :

### 1.2.1. Déplacement des pièces :

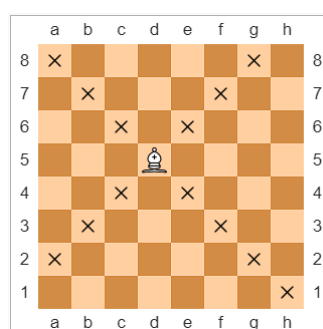
Les pions blancs peuvent se déplacer d'une seule case dans la ligne d'une coordonnées supérieure a la leur. (et de deux lorsque ils sont a la ligne 2).



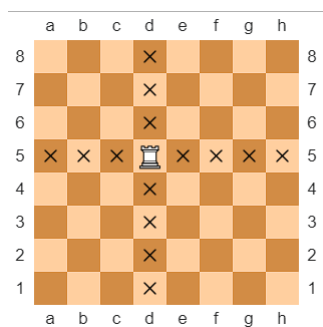
Les pions noirs peuvent se déplacer d'une seule case dans une ligne de coordonnées inférieure à la leur. (et de deux lorsque ils sont à la ligne 7).



Le fou peut se déplacer uniquement en diagonales

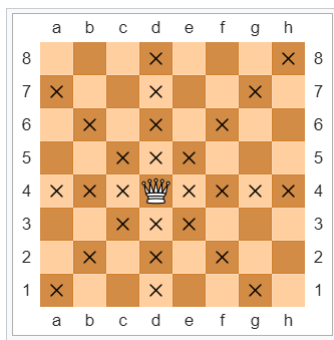


La tour peut se déplacer uniquement dans sa ligne ou sa colonne respective

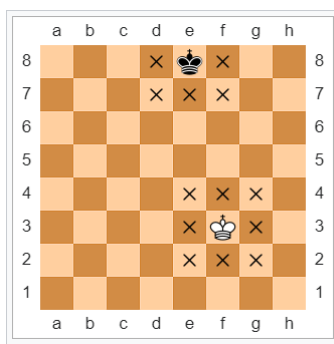


## PROGET 1

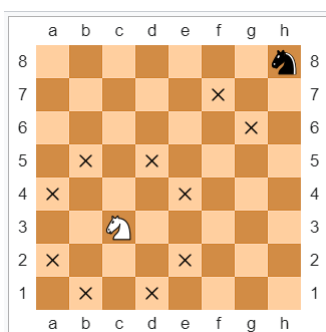
La dame quand a elle peut se déplacer en diagonales et dans ce ligne ou sa colonne respective.



Le roi peut se déplacer que d'une case mais dans tous les sens et toutes les directions.



Le cavalier peut se déplacer de deux cases dans toutes les directions et tous les sens, sauf la direction dans sa colonne respective et celle dans sa ligne respective et il, ne peut pas être bloqué par les pièces autour de lui car il saute par dessus.



A noter : que les pieces ne peuvent pas se déplacer lorsque il y a des obstacles autour deux a l'exception du cavalier qui peut sauter par-dessus, (mais il est bloqué dans le cas ou un obstacle se situe a l'endroit ou il peu atterrir).

### 1.2.2. Comment les pieces capturent les pièces des adversaires :

L'ensemble des pieces peuvent capturer les pieces ennemi etant sur leur trajectoir (bien en entendu elles ne peuvent avancé au dela du peces qu'elles capturent), a l'exception des pions qui eux ne peuvent manger que dans les diagonales (donc par la même direction que leur déplacement) mais par contre dans le même sens que celui pour se déplacer.

## PROGET 1

## 1.2.3. Règles spécifiques :

La promotion, vous pouvez récupérer a pièce de votre choix parmi les pièces capturées par votre adversaire si vous arrivez a atteindre la paroi opposée de l'échiquier avec une pièce quelconque.

L'échec, vous êtes en échec si votre roi est menacé d'être capturé au prochain tour mais qu'il à la possibilité d'échapper a son sort.

Roque, un roque est un coup spécial dans lequel le roi se déplace de deux cases vers la tour et la tour se déplace de l'autre cotés du roi. Un roque peut se dérouler lorsque la tour et le roi sont sur la alignés sur la ligne 1 (pour les pieces blanches) et sur la ligne 8 (pour les pieces noires).

(c'est un petit roque lorsque la tour se déplace de deux cases et un grand roque lorsque elle se déplace de trois cases).

Lorsque un est possible (

## 1.2.4. Fin de partie

Il y a plusieurs moyens de finir une partie :

Echec et mat :

L'échec et mat est une situation dans laquelle le roi est dans une situation dans laquelle il est menacé d'être capturé au prochain tour et qu'il n'a aucun moyen d'échapper a cette situation. Ainsi la partie s'arrête et l'adversaire est considéré comme vainqueur.

Pat :

C'est le cas dans lequel au moins un des participant ne jouer aucun coups et que son roi n'est pas en échec. Dans ce cas précis la partie s'arrête et la partie est déclarées nulle, il n'y a donc aucun vainqueur.

Abandon d'un joueur :

Lorsque un joueur quitte la partie la partie s'arrête automatiquement, la partie est déclarée nulle, il n'y a ainsi aucun vainqueur.

Perte au temps :

Si un des deux joueur ne joue pas un coup dans le temps imparti, il perds pour dépassement de temps, ainsi la partir se termine.

Autres cas (menant a des parties nulles) :

→Si une même position avec le même joueur ayant le trait se répète 3 fois.

→Si une même position (au même trait) se répète 5 fois.

→Automatiquement au bout de 75 coups après la capture ou la dernière poussée d'un pion.

→Impossibilité de mater

## 1.2.5. Notation de la partie :

Les coups joués sont notés afin de permettre l'arbitrage l'archivage de la partie, ce jeux utilisera la notation algébrique, ainsi.

## PROGET 1

Chaque colonnes de l'échiquier est notée par une lettre (allant de A à H) et chaque lignes de l'échiquier est notée par un chiffre (allant de 1 à 8). Ainsi chaque une des 64 cases se situe à des coordonnées cartésiennes en deux dimensions se composant d'une lettre et d'un chiffre.

Chaque pièces sont exprimée par leur initiale (R pour le Roi, D pour la Dame, T pour la Tour, F pour le Four, C pour le Cavalier et P pour le pion).

Ainsi chaque coups sont enregistrés :

→le numéro du coup (un  $\mathbb{N}$  allant de 1 à  $+\infty$ )

→l'initiale de la pièce jouée

→les coordonnées cartésiennes de la case d'arrivée.

Lorsque une pièce capture une pièce adverse l'on ajoute un x entre l'initiale de la pièce et la case d'arrivée

⇒Notation des coups spéciaux :

→Le roque est noté 0-0 pour le petit roque et 0-0-0 pour le grand roque.

→La prise en passant est notée comme une prise normale avec l'ajout d'un e.p a la fin de la note

→La promotion se note en ajoutant d'un = et de l'initiale de la pièce promue a la fin.

→Lorsque le roi adverse se trouve en échec, on ajoute un + à la fin de la note.

→si le roi est échec et mat on ajoute un + après le + de l'échec.

A noter: pour la notation j'ai décider de mettre l'initiale du pion pour avoir des trames de tailles égales (en dehors des cas spécifiques (capture/coups spéciaux)) afin de simplifier e code.

## 2. Objectifs secondaires :

### 2.1. Authentification :

Ajouter un système de comptes utilisateurs pour les joueurs, (ce compte servira à sauvegarder leurs divers ainsi que l'historique de leurs parties).

### 2.2. Scorage :

Ajouter un système de point de victoire, chaque joueur qui gagne une partie, ce voit attribué un point de victoire supplémentaire, (les joueurs seront aussi classés en fonction de leur nombre de points de victoires) et le classement est visible dans un onglet classement ajouter au menu du jeu.

### 2.3. Points d'XP :

Des points d'xp sont attribués aux joueurs à chaque parties, +10xp à chaque coups joués par l'adversaire, +20xp si l'on arrive à se sortir d'un échec et +1000xp si l'on arrive à gagner une partie (+1000xp supplémentaires si la victoire est faite par échec et mat).

Dans l'onglet classement, en plus du classement au nombre de victoire il y a aussi un onglet par rapport au nombre de points d'xp.

### 2.4. Note de bienséance :

A chaque fois que vous faites une partie d'échec, si vous la faites jusqu'à bout (sans abandonné, et sans écouler tout votre temps), vous gagnez 100 points de bienséance, votre note de bienséance sera calculée en divisant votre nombre de points de bienséance par rapport au nombre de partie d'échec que vous avez faite, ainsi cela vous permettra d'avoir une note /100 de votre bienséance (qui sera affichée sur l'onglet de votre partie).

### 2.5. Mode spectateur :

Rajouter la possibilité de se connecter en tant que spectateur à une partie en cours et de pouvoir regarder la partie d'échec se jouer.

### 2.6. Protection anti-tcheat :

La protection anti cheat sera une fonction de mon programme qui vérifiera que les coups joués par l'adversaire respectent bien les règles des échecs (et que donc le joueur en face n'a pas tcheaté son jeu pour tricher). En cas de détection de tcheat l'anti cheat met fin à la partie (la considère comme partie nulle) et enlève 100 points de bienséance au tcheateur, et si un joueur tcheate 3 fois, son compte en ligne est supprimé par l'antitcheat.



### 3. Quelques approfondissements Techniques :

#### 3.1. Multi-Threading :

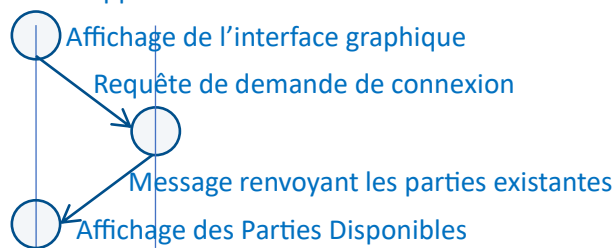
L'application disposera de plusieurs Threads :

- Un thread pour l'interface homme machine
- Un thread pour la gestion des règles du jeu
- Un thread pour les échanges http avec le serveur distant
- Des thread pour la gestion du temps.

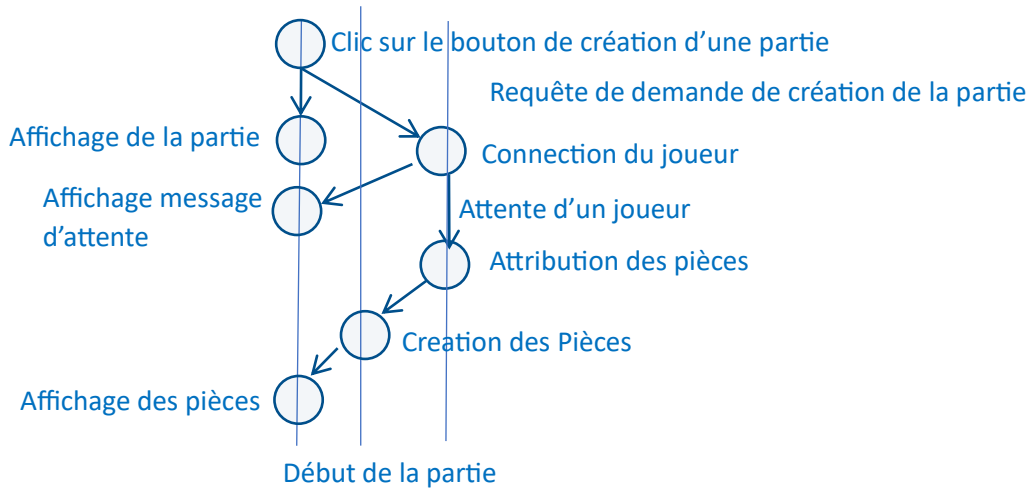
Il y aura divers interactions entre les Threads :

##### 3.1.1. Lancement de L'application :

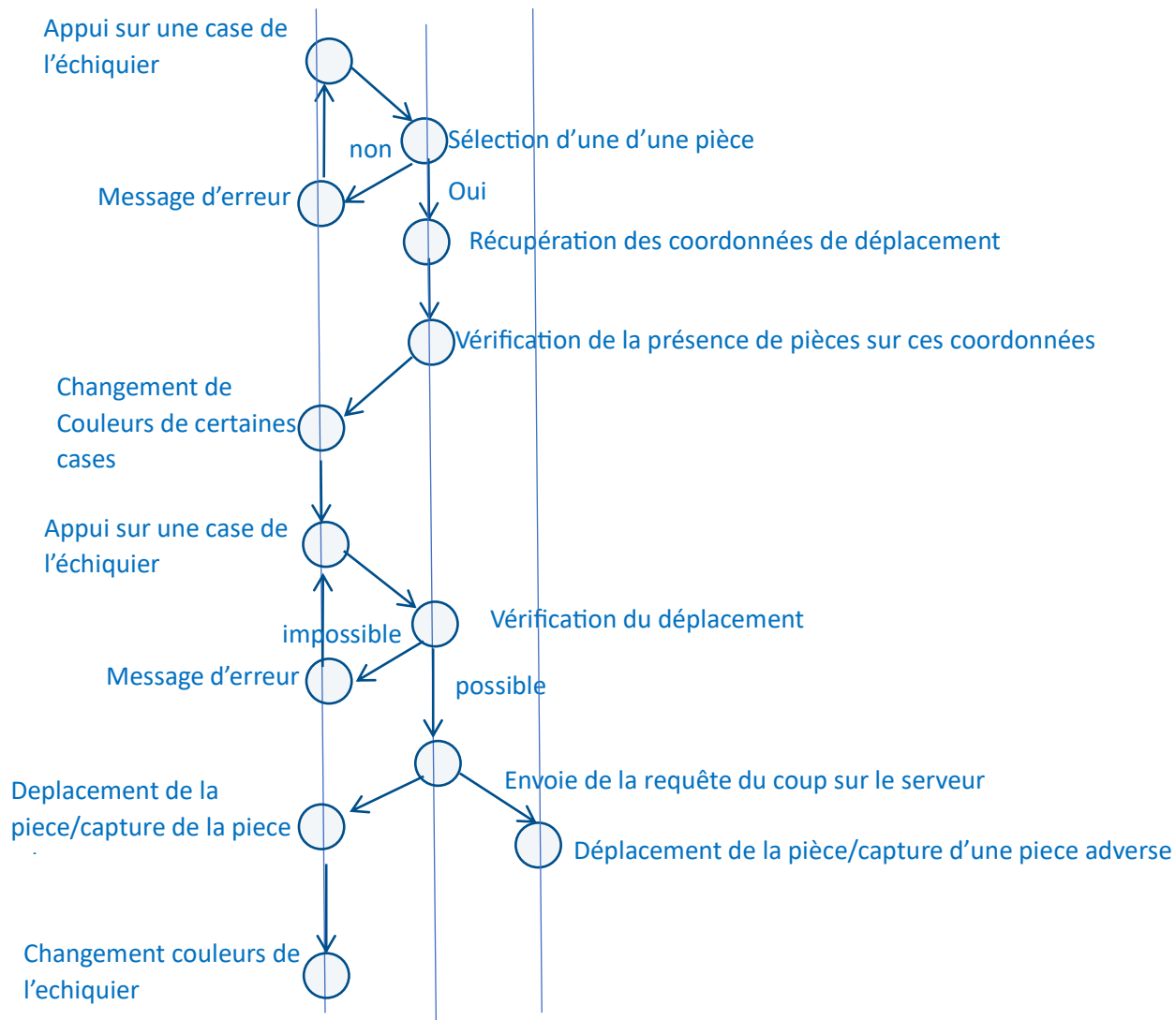
Ouverture de l'application



##### 3.1.2. Creation d'une partie :



### 3.1.3. Lorsque vous jouez un coups :



### 3.2. Programmation Orientée Objets :

#### 3.2.1. Objet cases

Les cases de l'échiquier seront des objets instanciés de la classe case :

Classe case comportant :

- des coordonnées
- une largeur (px)
- une longueur (px)
- une marge supérieure (px)
- une marge latérale (px)
- une couleur RVB
- une possible image affichée

La classe cases disposera de divers fonctions :

- CreateCase(coordonnées, largeur, marge supérieure, marge latérale, couleur)
- SetImage(coordonnées, image) (affichera l'image si les coordonnées correspondent).
- SetColor(coordonnées, couleur) (changera la couleur de la case si les coordonnées correspondent)

#### 3.2.2. Object Pieces :

Les pièces d'échecs seront des objets instanciés de la classe pièce :

Classe pièce comportant :

- un id (unique)
- un nom
- une « apparence » (lien vers une image)
- des coordonnées
- une orientation (nord ou sud)
- un type de déplacement (char, move or jump)
- un ou plusieurs sens de déplacements (TOP, BOTTOM, LEFT, RIGHT, LEFT BOTTOM, RIGHT BOTTOM, LEFT TOP, RIGHT BOTTOM).
- une distance de déplacement minimale (int)
- une distance de déplacement maximale (int, (-1 si illimitée))
- un ou plusieurs sens de capture (TOP, BOTTOM, LEFT, RIGHT, LEFT BOTTOM, RIGHT BOTTOM, LEFT TOP, RIGHT BOTTOM).
- une distance de capture minimale (int)

## PROGET 1

→une distance de capture maximale (int, (-1 si illimitée))

La classe pièce disposera de divers fonctions :

→CreatePiece(nom, apparence, coordonnées, orientation, type de déplacement, sens de déplacements, distance de déplacement minimale, distance de déplacement maximale, sens de capture, distance de capture minimale, distance de capture maximale); qui retourne l'objet créé.

→SelectPiece(coordonnées); qui retournera les coordonnées des cases sur lesquelles la piece peut se déplacer et celles ou la piece peut en capturer d'autre. Si il n'y as pas de pièce aux coordonnées entrées la fonction retourne 0.

→MovePiece (coordonnées) ; qui retournera la trame du coup si le mouvement est permis.

→PieceCapture(coordonnées) ; qui retournera la trame du coup si la capture est permise.

→GetPieceID(coordonnées); qui retournera l'id de la piece si une pièce est présente aux coordonnées données et 0 si il n'y as pas de pieces a cette coordonnées

→GetPieceLink(coordonnées); qui retournera le lien de l'image d'apparence de la piece si une pièce est présente aux coordonnées données et 0 si il n'y as pas de pieces a cette coordonnées

→SetPiecePosition(coordonnées) ; qui permet de donner de nouvelles coordonnées a une pieces (donner les coordonnées 0-0 pour mettre une pièce capturée) (retournera 1

### 3.3. Interactions entre L'ihm, le serveur et la classe Pièce