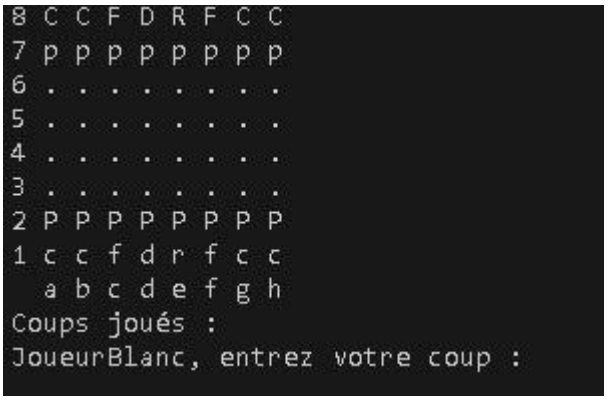
PROJET JEUX D'ÉCHECS :



Instructions de lancement :

Pour exécuter le programme Java, voici les étapes à suivre à partir du terminal :

1. Se positionner dans le dossier contenant les fichiers .java : cd SAE_Java_Echecs_AMMOUR_MESLIN
2. Compiler tous les fichiers java : javac *.java
3. Lancer le programme via la classe principale (ici, Main) : java Main

Bilan :

Ce qu'on a fait

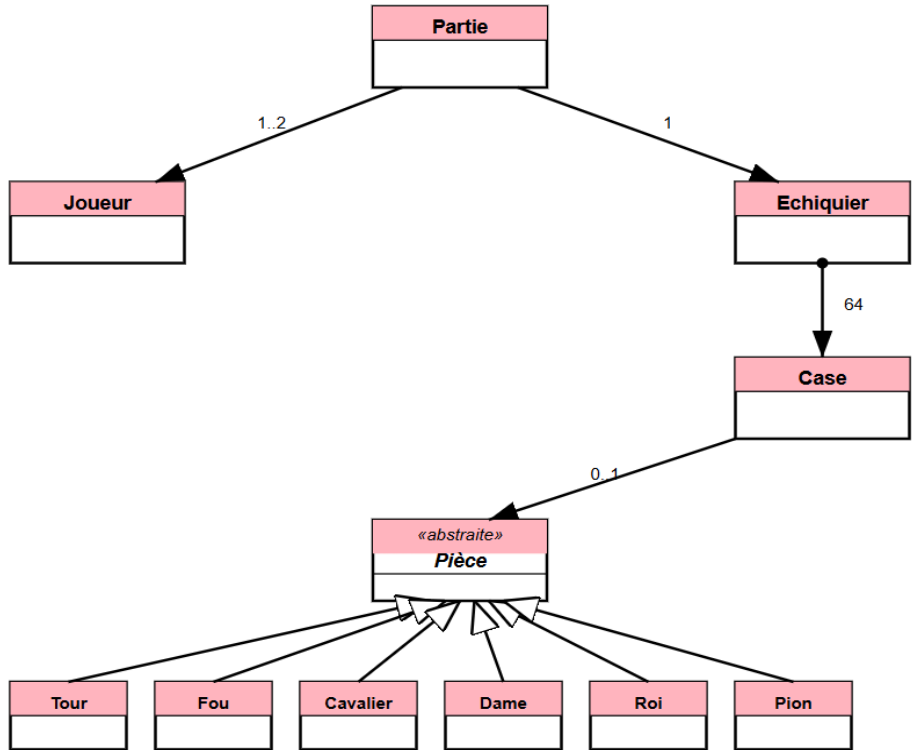
On a réussi à créer un jeu d'échecs entre deux joueurs humains. Le programme demande les noms, affiche l'échiquier en texte, garde l'historique des coups, et vérifie si les coups sont valides selon les règles. Il détecte la fin de partie (mat) et annonce le gagnant. On a bien géré les pièces suivantes : Tour, Fou, Dame, Roi et Pion.
L'interface est simple à utiliser. Le code est bien organisé, avec des classes claires, bien séparées, et il respecte les règles de la programmation orientée objet. Les diagrammes UML demandés ont été faits.

Ce qu'on n'a pas fait

Le Cavalier n'a pas été codé : son déplacement en "L" était plus compliqué. On n'a pas non plus fait la gestion du temps (horloge) car c'était optionnel et ça aurait été trop technique pour ce projet.
On n'a pas ajouté non plus le roque, la promotion du pion ou la prise en passant, car ce n'était pas demandé dans la version simplifiée du jeu.

Ce qui fonctionne

Le jeu fonctionne bien du début à la fin. Les pièces se déplacent comme il faut, l'affichage et l'historique sont mîs à jour, et la fin de partie est bien reconnue.



Légendes :

flèches pleines : associations | Triangle vide : héritage | Triangle noir : composition

Justification :

Ce diagramme UML représente la structure générale d'un jeu d'échecs en identifiant les principales classes nécessaires au fonctionnement du système. On retrouve la classe Partie qui gère l'ensemble du jeu, les deux Joueurs qui participent à la partie, l'Echiquier qui constitue le plateau de jeu, les 64 Cases qui composent cet échiquier, et enfin toutes les Pièces avec leurs spécialisations (Tour, Fou, Cavalier, Dame, Roi, Pion).

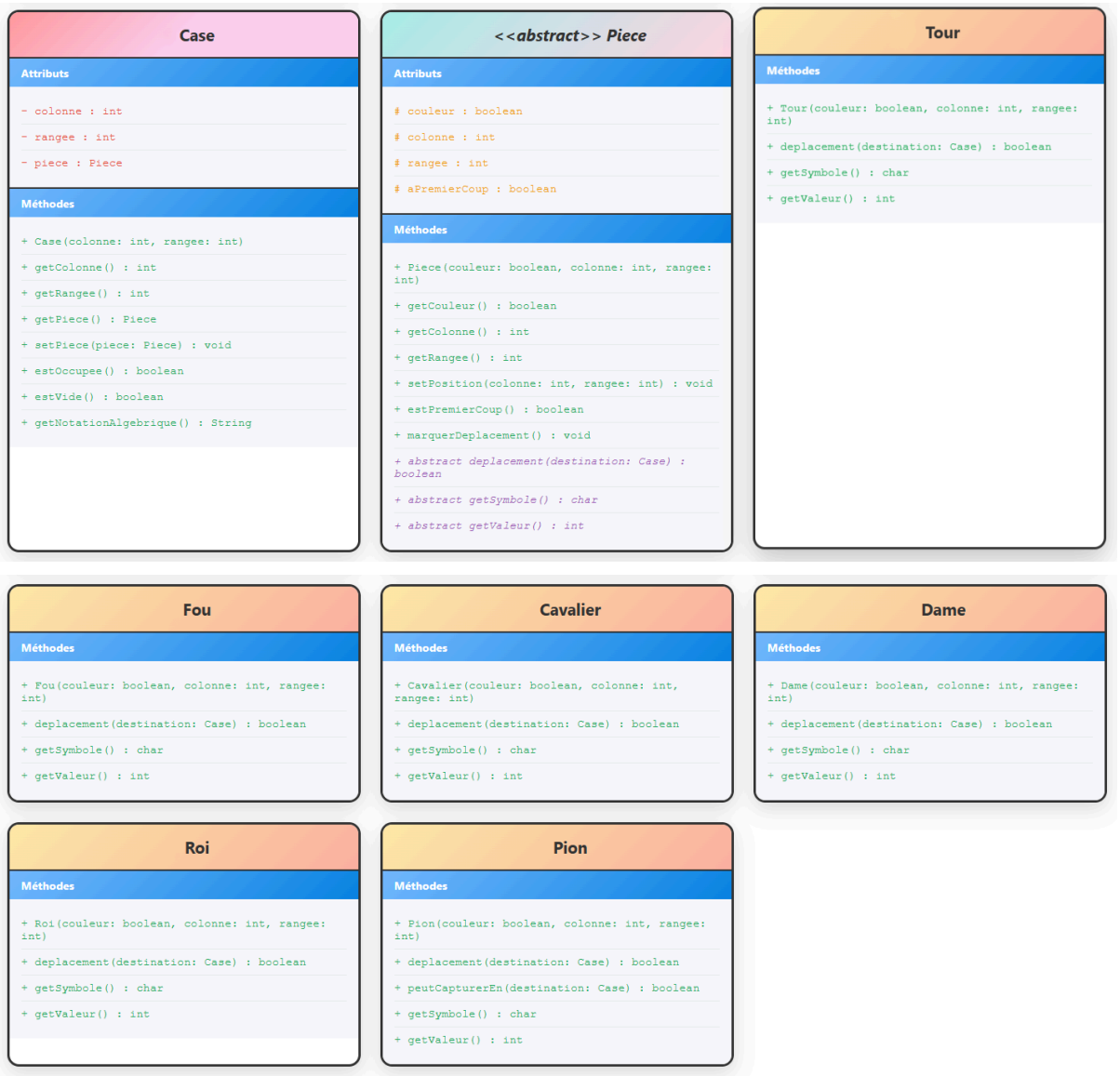
Les parties inférieures des rectangles de classe sont volontairement laissées vides car cette première étape de modélisation se concentre uniquement sur l'identification des classes et leurs relations. Selon la méthode demandée, il faut d'abord établir la structure globale avant de détailler les attributs et méthodes de chaque classe.

Les chiffres présents sur le diagramme indiquent les multiplicités des associations ou combien d'instances d'une classe peuvent être liées à une autre. Par exemple, "1..2" entre Partie et Joueur signifie qu'une partie peut avoir un ou deux joueurs, "64" entre Échiquier et

Case indique qu'un échiquier contient exactement 64 cases (8 x 8), et "0..1" entre Case et Pièce montre qu'une case peut être vide ou contenir une seule pièce.

Dans notre diagramme, trois types de relations sont utilisés : les flèches pleines représentent les associations simples entre classes (comme Partie-Joueur), les triangles vides indiquent l'héritage où une classe fille hérite des propriétés de sa classe mère (toutes les pièces héritent de la classe Pièce), et le triangle noir symbolise une composition forte, voulant dire que si l'élément parent disparaît, les éléments enfants n'ont plus de raison d'exister (si on détruit l'échiquier, les cases n'existent plus).

UML 2 :



- La classe Partie contrôle le jeu. Elle contient les deux joueurs, l'échiquier, et garde en mémoire tous les coups joués dans historiqueCoups. Ses méthodes permettent de démarrer une partie, faire jouer les coups, et vérifier si la partie est terminée (échec et mat).
- La classe Joueur représente chaque joueur avec son nom et sa couleur (blanc ou noir). Elle gère aussi le temps de jeu avec des méthodes pour compter le temps total et le temps par coup.
- L'Echiquier représente le plateau de jeu. Il contient 64 cases et permet de placer les pièces, les déplacer, et vérifier si un chemin est libre pour un déplacement.
- Chaque Case a une position (colonne et rangée) et peut contenir une pièce. Elle permet de savoir si une case est occupée et donne sa position en notation d'échecs.
- Toutes les pièces héritent de la classe Piece qui contient leur couleur, position, et si c'est leur premier coup. Chaque type de pièce (Tour, Fou, Cavalier, Dame, Roi, Pion)

a ses propres règles de déplacement dans sa méthode deplacement(). Le Pion a une méthode spéciale peutCapturerEn() pour la prise en passant.

Relations entre les Classes

Une Partie a exactement 2 Joueur et 1 Echiquier. L'Echiquier contient 64 Case, et chaque Case peut avoir 0 ou 1 Piece. Toutes les pièces (Tour, Fou, Cavalier, Dame, Roi, Pion) héritent de Piece, ce qui permet de les traiter de la même façon dans le code en gardant leurs règles particulières.