

# Majeure Informatique: Jeu d'échecs

BOYER Timothé, MOURET Basile, HACINI Malik

26 Septembre 2023

## Table des Matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Conception</b>	<b>1</b>
2.1	Règles traitées . . . . .	1
2.2	Architecture . . . . .	2
2.2.1	Diagramme de classe UML . . . . .	2
<b>3</b>	<b>Conception</b>	<b>2</b>
3.1	Conception du Jeu d'échecs . . . . .	2
3.1.1	Classe Piece . . . . .	2
3.1.2	Classe EtatJeu . . . . .	2
3.1.3	Classe Joueurs . . . . .	3
3.2	IA . . . . .	3
3.3	Le Programme main . . . . .	3
<b>4</b>	<b>Tests</b>	<b>3</b>
<b>5</b>	<b>peut être un de plus jsp</b>	<b>3</b>

## 1 Introduction

L'objectif de ce projet est de programmer (et tester) un jeu d'échecs qui permet à deux joueurs de s'affronter, chaque joueur peut être soit un humain, soit l'ordinateur via une IA (intelligence artificielle) . Le jeu se jouera dans un terminal, et si les deux joueurs sont humains, ils utiliseront le même clavier.

## 2 Conception

### 2.1 Règles traitées

traitées: promotion, roque non traitées : en passant

## 2.2 Architecture

Ce projet a été réalisé en utilisant les outils de la programmation orientée objet (POO). Les différentes composantes d'un jeu d'échecs (joueurs, pieces) sont donc représentées par des classes.

### 2.2.1 Diagramme de classe UML

a mettre

## 3 Conception

Il y eu 2 grandes phases de développement du projet : La conception du jeu entre humains et la mise en place de l'IA. Chaque phase à donné lieu à plusieurs algorithmes principaux que nous allons détailler.

### 3.1 Conception du Jeu d'échecs

Nous avons commencé par créer en parallèle les 3 classes constituant le jeu: Piece, EtatJeu et Joueurs. Ces classes sont toutes associées (CF 2.2.1), donc nous ne les avons pas réellement rédigés indépendamment.

#### 3.1.1 Classe Piece

La classe Piece et ses sous-classes sont les lieux où les règles du jeu sont définies. On représente chaque type de pièce par une sous-classe de la classe Piece. Chacune de ses pièces possède sa propre méthode (override) coups-possibles, qui encode les règles du jeu qui lui sont relatives. Ces méthodes nécessitent évidemment d'avoir accès à l'état du jeu.

Enfin, la classe Piece possède une méthode coups-légaux (donc identique pour tout les types de pièces) qui trie une liste de coups possibles en enlevant ceux qui mettent en échec le roi. Ces méthodes nécessitent évidemment d'avoir accès à l'état du jeu, que nous avons donc défini par la suite.

---

```
1
2  #PLACEHOLDER
```

---

#### 3.1.2 Classe EtatJeu

Nous avons d'abord mis en place le plateau, ainsi que l'affichage et la sauvegarde de celui-ci.

---

```
1
2  #PLACEHOLDER
```

---

Plusieurs choix importants ont été réalisés pour la mise en place du plateau : Nous le représentons via un dictionnaire, contenant des coordonnées (tuple) en clé et des pièces (objets de la classe

Piece) en valeur. Cette représentation possède plusieurs avantages : Il est facile d'accéder au symbole UTF-8 des pièces pour l'affichage, à la couleur d'une pièce, ou même à ses coups possibles.

Aussi, un objet EtatJeu possède une liste de deux listes de pièces, les blanches et les noires. L'accès aux pièces noires ou blanches est réalisé en accédant au bon indice de la liste pieces via le trait.

Nous avons ensuite ajouté les fonctions relatives à la vérification de l'état du jeu (échec, échec et mat, égalité).

---

```
1
2  #PLACEHOLDER
```

---

### 3.1.3 Classe Joueurs

C'est la classe Joueurs qui sera l'interface entre le programme main et les classes Piece et EtatJeu, via la méthode jouer-coup. En prévision de l'implémentation de l'IA, on implémente une sous classe Humain, qui a sa propre méthode jouer-coup, car celle de l'IA sera différente.

---

```
1
2  #PLACEHOLDER
```

---

## 3.2 IA

A FAIRE

## 3.3 Le Programme main

Au final, le jeu doit se jouer depuis un simple script main, qui utilise nos classes définies dans des fichiers annexes. On définit alors une fonction partie, qui crée (à partir des choix de l'utilisateur) et fait jouer une partie à l'utilisateur. On place ensuite cette fonction dans une boucle, et le jeu est prêt.

---

```
1
2  #PLACEHOLDER
```

---

## 4 Tests

## 5 peut être un de plus jsp