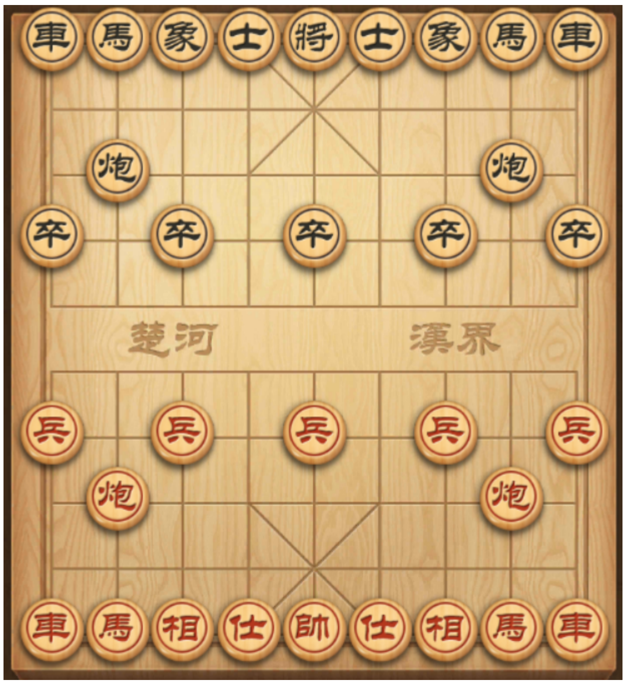
Xiangqi

I. Introduction

In this project, you are required to implement the classic board game **Xiangqi** using the **Java programming language**.



Xiangqi is a traditional two-player strategy game with a history of more than a thousand years. It emphasizes **tactical planning, spatial reasoning, and prediction of the opponent's moves**. The objective of the game is to **checkmate the opponent's King** by strategically maneuvering your pieces while protecting your own.

II. Game Interface & Basic Elements

The game interface consists of a **9×10 board** separated by a river in the middle. Each player controls 16 pieces of red or black color, arranged symmetrically. The pieces include:

- **King (帅/将)**
- **Advisors (仕/士)**
- **Elephants (相/象)**
- **Horses (马)**
- **Rooks(车)**
- **Cannons (炮)**
- **Soldiers (兵/卒)**

Each piece follows unique movement rules. The interface must display the **board**, **pieces**, **turn indicator**, **move record**, and **status area** (e.g., check 将军, checkmate 将死, draw 平局).

III. Detailed Game Rules

1. **Turn-based Play**
    - Two players (Red and Black) alternate turns.
    - Red always moves first.

      Each player can **move only one piece per turn, and must move a piece**.
2. **Movement Rules** （Appendix 3）
    - Each type of piece has fixed movement patterns.
    - Pieces cannot move outside their legal area (e.g., Advisors inside the palace, Elephants cannot cross the river).
    - Cannons capture by leaping over exactly one piece.
3. **Capturing and Checking**
    - A piece can capture(吃) an opponent’s piece by moving into its square.
    - “Check(将军)” occurs when the King is threatened by capture in the next move.
4. **Winning and Draw Conditions**
    - The game ends when one player’s King is checkmated. The **basic** part of the project only requires the completion of the victory/defeat logic.
    - The **draw** will be regarded as a **bonus**. The specific rules can be found in the appendix.

IV. Project Requirements

This project consists of **six major tasks**. The total score is **100 points**, distributed as follows:

Task 1: Game Initialization (10 points)

1. The game should correctly display the **board layout**, including river, palace, and initial piece positions.
2. When starting a new game, the board resets to its default initial state.
3. The interface should indicate which player’s turn it is.
4. Players can **restart** a new game at any time without restarting the program.

Task 2: Multi-user Login (15 points)

1. Implement a login system supporting both **guests** and **registered** users.
2. Guests can play directly but cannot save or load game progress.
3. The user login interface includes a registration page and allows login after entering account credentials.
4. After the program exits and is run again, previously registered users can still log in.

Task 3: Save and Load Games (15 points)

1. Each registered user can **save** their current game progress (board state, player turn, move history).
2. Each user must have **at least one save slot**(multiple saves are optional but not required).
3. Players can **load** their previous save from the start menu.

4. Handle corrupted or modified save files gracefully (do not crash).
5. Implementing **auto-save on exit** or **time-stamped saves** earns extra credit.

Task 4: Gameplay Logic (30 points)

1. Implement **legal move rules** for all 7 types of pieces. The specific rules can be found in the appendix.
2. Ensure that illegal moves (e.g., Elephant crossing river, face-to-face Generals) are prevented.
3. Implement **capturing, checking, and checkmate detection**.
4. Show appropriate messages for **check**, **checkmate**, **stalemate**, and **draw**.
5. Provide an **option to surrender** (resign).
6. Display **the previous move** on the interface (e.g., highlight the moved piece or show textual record).
7. **Detect game victory and display a victory message.**
8. Implementing full **draw** and **stalemate** rule detection is complex — count it as a bonus. (不强制要求)

Task 5: Graphical User Interface (GUI) (10 points)

1. Implement a graphical board and interactive piece movement using **JavaFX**, **Swing**, or any other **Java GUI framework**.
2. Players should be able to move pieces **via mouse click-and-drop** or **by selecting start and destination tiles**.
3. Display game status (turn, check, game over) visually.
4. If your program runs only via command line without GUI interaction, you will not get full marks in this section.
5. Independently creating the GUI without using the provided demo code will count as advanced points.

Task 6: Advanced Features (20 points)

You may gain additional points by implementing **advanced or creative features**, including but not limited to:

1. AI opponent with adjustable difficulty levels （人机对战）
2. Move suggestions or hint system （下一步提示）
3. Timer and time-limited mode （不同模式）
4. Game replay (move-by-move playback) （游戏整局回放）
5. Move record export/import (PGN-like format) （残局导入）
6. Enhanced graphics and piece animations （更好的图形和动画）
7. Online multiplayer or LAN play （联网对战）
8. Sound effects and background music （背景音乐）
9. Support for different board/piece themes （不同主题界面）
10. Highlight legal moves indicators （合法移动显示）
11. Players can **undo** the move（游戏中悔棋）

V. Appendix

1. **Initial Setup Example**

```
  1 2 3 4 5 6 7 8 9
┌───────────────────┐     黑
│ R N B A K A B N R │
│ . . . . . . . . . │
│ . C . . . . . C . │
│ P . P . P . P . P │
│ . . . . . . . . . │
│ . . . . . . . . . │
│ p . p . p . p . p │
│ . c . . . . . c . │
│ . . . . . . . . . │
│ r n b a k a b n r │
└───────────────────┘     红
  9 8 7 6 5 4 3 2 1
```

2. **Notation Examples**

英文符号	中文对应	英文和含义
R	车	Rook
H	马	Horse
C	炮	Cannon
A	士 / 仕	Advisor
E	象 / 相	Elephant
K	将 / 帅	King
P	兵 / 卒	Soldier
+	进	advance / move forward
-	退	retreat / move backward
=	平	move horizontally / move sideways
数字	路数（纵列）	Each player’s file numbering (1–9) is counted from their own right to left

- `R2+6`: Rooks from file 2 advances 6 ranks.
  - 中文翻译：“车二进六”
  - 操作说明： **“R”** 表示 **Rook（车）**，**2** 表示该车原本在**第2路（从红方右数第二列）**，**“+6”** 表示 **向前（进）移动6格**

    即：红方右边的车向前走6步。
- `C8=5`: Cannon from file 8 moves horizontally to file 5.
  - 中文翻译： “炮八平五”

  - 操作说明：**“C”** 表示 **Cannon（炮）**，**“8”** 表示原来在**第8路（从红方右数第八列）**，**“=5”** 表示**平到第5路（横向移动到中线）**

    即：右侧的炮横向平移到中线位置。

3. **中国象棋棋子移动规则简表（Xiangqi Piece Movement Summary）**

棋子	**英文名称**	**简要移动规则（中文）**
帅 / 将	King	**九宫内走一格直线；不得对面见将**
仕 / 士	Advisor	**九宫内走斜线一格（对角线）**
相 / 象	Elephant	**走田字（斜线两格）；不能过河；堵象眼不可走**
马	Horse	**走日字（先直一格再斜一格）；蹩马腿不可走**
车	Rook	**直线任意步；无障碍可行**
炮	Cannon	**平直走；吃子需隔一子（炮架）**
兵 / 卒	Soldier	**未过河只能直进一格；过河后可左右平移一格；不能后退**

We provide a more comprehensive English introduction to the rules of piece movement. Please download and check the "*Tutorial.jpeg*" file. (this content is from https://www.xiangqi.com/)

4. **Draw and Stalemate Rules**

In Xiangqi, the game may end **without a winner** under certain circumstances. While both result in a tie, **draw** and **stalemate** have different meanings and detection rules.

Draw (平局)

A **draw** occurs when neither player can achieve victory or when the rules dictate a tie.
Draws can happen in several ways:

- **Mutual agreement** — both players agree to end the game as a tie.
- **Threefold repetition** — the same board position (same pieces and same player to move) appears **three times**.
- **Perpetual check or chase** — one player repeatedly checks or attacks without progress.
- **Dead position** — neither side has sufficient material or opportunity to checkmate.
- **Stalemate** — a specific type of draw (see below).

When a draw is detected or agreed upon, the game ends immediately with **no winner or loser**.

Stalemate (困毙)

A **stalemate** occurs when the player whose turn it is to move **is not in check**,
but **has no legal moves available** — any move would place their own King in check.

In this case:

- The game ends immediately.
- The result is considered a **draw** (tie).
- It usually happens in the endgame when a player’s pieces are completely blocked.

Example:

> If it’s Black’s turn, the Black King is safe but every possible move is illegal
> (e.g., moving any piece would expose the King to check),
> the game ends as a **stalemate**.