# Chinese Chess Data Format (En)

(Designed for CS102A Fall 2019 Final Project)

Current version: v2.4 (Updated on 11/24)

## Abstract

This document describes a file format for storing chessboards and moving sequences.

When writing the *Xiangqi (Chinese Chess)* program, please follow the format standard so that data can be transferred among different implementations, and student assistants can test the program.

Requirements for encoding and linebreaks: All files should be stored in **UTF-8** format, and all linebreak characters should be `\n` (UNIX linebreak format).

## Denoting pieces

Use a single English character to represent each piece. Black pieces are denoted in upper case, while red pieces in lower case. Details are shown in the chart below:

| Piece | English name | Black player (upper case) | Red player (lower case) |
|---|---|---|---|
| 將 / 帥 | [G]eneral | G | g |
| 士 / 仕 | [A]dvisor | A | a |
| 象 / 相 | [E]lephant | E | e |
| 馬 / 傌 | [H]orse | H | h |
| 車 / 俥 | [C]hariot | C | c |
| 砲 / 炮 | Ca[N]non | N | n |
| 卒 / 兵 | [S]oldier | S | s |

## Metadata

Metadata can be stored at the beginning of the file to provide more information.

1. Each line of metadata begins with an `@` character, following the format of key-value pairs such as: `@[key]=[value]`;
2. The metadata field ends with a line `@@`;
3. The main content should start in a new line after the metadata field.

## Comments

A comment begins with a `#` character, and ends at the end of the line.

Comments should be ignored when resolving data.

For example:

```
--------- # Inline comment: This is the River.
@LAST_MOVER=BLACK # Comment in metadata: Last mover is the black player.
# This is a comment which spans a whole line.
```

## Chessboards (.chessboard)

Store the chessboard mainly according to the actual situation so that human can read directly.

### Metadata

1. Essential metadata: `LAST_MOVER` indicating the player who took the last step. The value should be `BLACK` or `RED`.
2. Optional metadata: One may add other metadata to fit the requirements of the program.

### Content

1. Denote by symbols shown in section ***Denoting pieces*** where pieces are located;
2. Denote by `.` (dot) where no pieces exist;
3. Use `---------` (9 dashes) to denote the River;
4. End the chessboard with a new line.

### Loading data

1. First read metadata from the chessboard file, and preprocess according to the requirements of the program;
2. Read the chessboard and verify the validity:

   For invalid cases, prompt the user and explain the problem, filename and line number, then stop loading. Possible invalid conditions are:

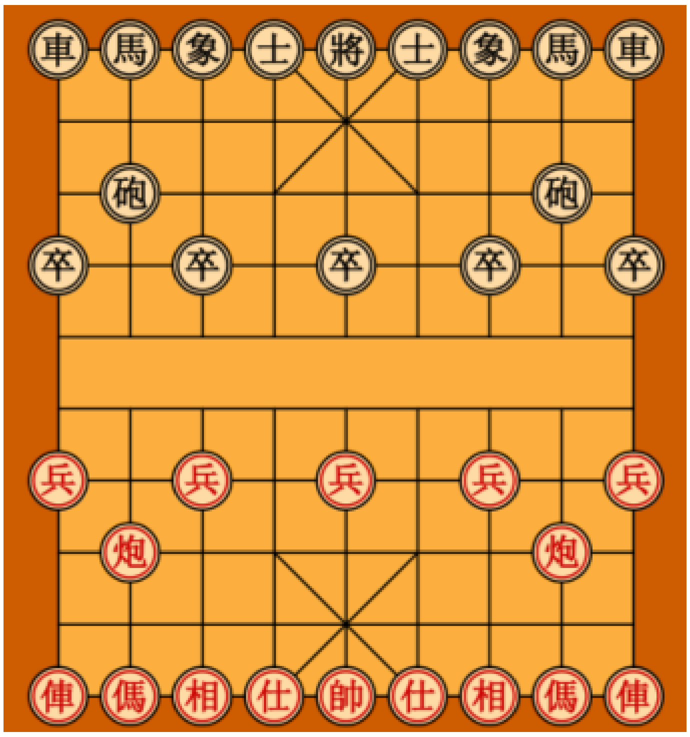
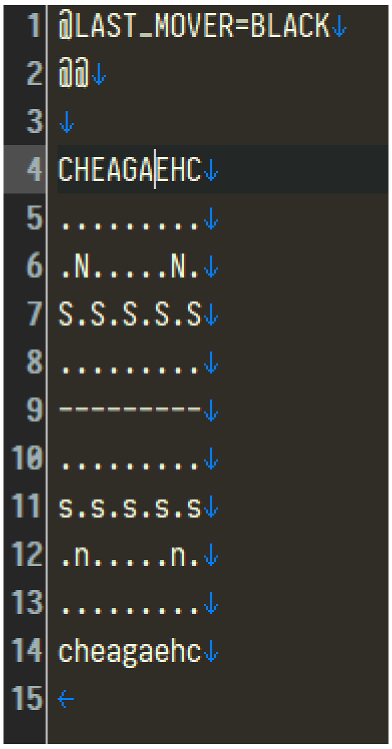
   a. *Invalid Dimension*: Wrong length or width;

   b. *River Missing*;

   c. *Invalid Chess Amount*: the number of pieces of a side / kind is greater than that at the opening;

   d. *Space Missing*.
3. If the chessboard is loaded, prompt the user that loading is successful (one may use pop-up windows, command-line output or GUI).

### Example

| The chessboard | Corresponding text |
|---|---|
|   車 馬 象 士 將 士 象 馬 車<br>砲 砲<br>卒 卒 卒 卒 卒<br>兵 兵 兵 兵 兵<br>炮 炮<br>俥 傌 相 仕 帥 仕 相 傌 俥 |  |

```
@LAST_MOVER=BLACK
@@

CHEAGAEHC
.........
.N.....N.
S.S.S.S.S
.........
---------
.........
s.s.s.s.s
.n.....n.
.........
cheagaehc

```

## Moving sequences (.chessmoveseq)

Store the sequence from top to bottom.

### Metadata

1. Essential metadata: `TOTAL_STEP` indicating the total number of lines storing the sequence.
2. Optional metadata: One may add other metadata to fit the requirements of the program.

### Content

1. Each line contains four integers: `[original x] [original y] [destiation x] [destination y]`, the corresponding piece of which should belong to the current player's own;
2. Origin coordinate: The left-bottom for black player; and the right-top for red player, denoted by `(1,1)`;
3. After all steps, end the file with a new line;
4. Parse to the end of the file by default. For invalid steps, skip the step and prompt the user about the position and content, then continue parsing.

### Loading data

The same as ***Chessboards* > *Loading data***.

Possible invalid conditions are:

a. *Position Out of Range*;

b. *Invalid From Position*: The original position contains no pieces / no pieces of the current player's / no pieces of the corresponding type;

c. *Invalid To Position*: The destination position contains a piece of the current player's;

d. *Invalid Move Pattern*.

## Example

**Step 1 (Red)** The Soldier at column 7 moves 1 step forward: 7 4 7 5

**Step 2 (Black)** The Cannon at column 2 moves horizontally to column 3: 2 3 3 3

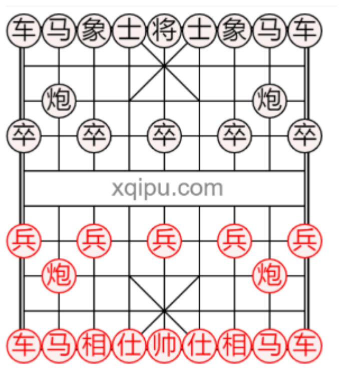
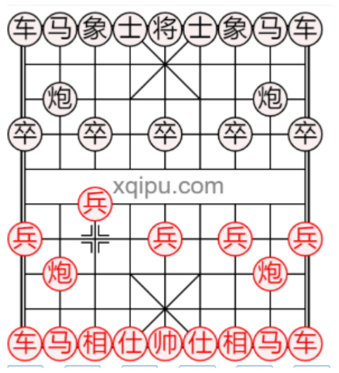
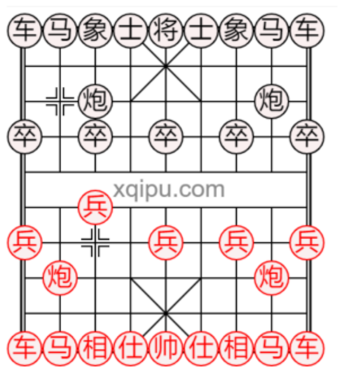
File content:

```
@TOTAL_STEP=2
@@

7 4 7 5
2 3 3 3

```

Corresponding chessboard:

| Original state | Step 1 | Step 2 |
|---|---|---|
|  |  |  |