Dark Chess

Dark chess (象棋暗棋，又名翻棋) is a variant of traditional Chinese chess (中国象棋), as shown in Figure 1. It is a board game for two players on a 8×4 grid, i.e., half of the traditional Chinese chess board. Dark chess is a social game, usually played for fun rather than serious competition.

Figure 1: An example of Dark chess.

Introduction

Dark chess uses the original Chinese chess pieces, in total 32 pieces, with 16 RED pieces and 16 BLACK pieces. The details of pieces are shown in Table 1 below. All 32 pieces are initially *shuffled* and *randomly* allocated *face-down* to squares on the board. Note that the pieces are placed inside the squares of the 8×4 grids, rather than on the intersections as in Chinese chess. During a game, two players take turns performing an action each turn.

There are several variations on the Dark chess rules. In this project, we will follow the rules specified in the next section. These rules are similar to the rules of Dark chess in the Tiantian Xiangqi app (天天象棋翻翻棋), *with an exception that all 32 pieces are face-down initially (while Tiantian Xiangqi has four pieces face-up initially)*.

Table 1: Details of the chess pieces.

Name of piece	Number of pieces each side	Points
General	帥 將 ×1	30
Advisor	仕 士 ×2	10
Minister	相 象 ×2	5
Chariot	俥 車 ×2	5
Horse	傌 馬 ×2	5
Soldier	兵 卒 ×5	1
Cannon	炮 砲 ×2	5

Rules

Please strictly follow the rules in this section for your program. If you have any questions about the rules, please ask the instructors immediately.

1. At the beginning, all 32 pieces are *shuffled* and *randomly* allocated *face-down* on the board. Note that the pieces are placed inside the squares of the 8×4 grid, rather than on the intersections as in Chinese chess.

2. The first player turns up a piece to begin the game. The color of that first uncovered piece is the color the first player will play in the game. The second player then makes a move, and the two players alternate until the game is finished.

3. When it is a player's turn, the player can perform three kinds of actions: turn a piece face-up, move a piece, or capture an enemy piece.

- **Turning over a piece**: Turning a piece from face-down to face-up is a legal move if there are any face-down pieces on the board. Once revealed, a piece may move, capture, or be captured.
- **Moving a piece**: A player may only move face-up pieces of their own color. A piece (*except Cannon*) may move *only one square* up, down, left, or right. A piece can never move onto a square that is already occupied unless such a move is a legal capture.
- **Capturing an opposing piece**: A player may use a their own face-up piece to capture a face-up piece of the opposing color. In all captures, the captured piece is removed from the board and its square is occupied by the capturing piece. The rules for piece capturing are as follows.
  - The pieces are ranked, forming a hierarchy with the General at the top and Soldiers at the bottom. The ranking of pieces goes as follows: General>Advisor>Minister>Chariot>Horse>Soldier. Only pieces of equal or lower rank may be captured, with *two exceptions:* i) General can only be captured by Soldier; ii) Cannon captures in a different way (more details below). Below are some legal and illegal example actions.
    - [LEGAL] A General captures a Minister. A Soldier captures a General. A Red Horse captures a black Horse (even they have equal rank).
    - [ILLEGAL] A Minister captures an Advisor.
  - All pieces capture exactly as they move: *one square* up, down, left, or right.
  - The Cannon is not included in the ranking because it is exceptional: it captures in an unusual way, *it can capture a piece of any rank (including General), and yet is vulnerable to be captured by any piece except the Soldier*. A Cannon captures in the same way as in Chinese chess: it needs to jump over exactly one intermediate piece (called a *screen*) along a single row or column of the board. The color of the screening piece does not matter---it can be the player's own piece or enemy's piece, or even face-down unrevealed piece.
  - The Cannon can capture a piece even if the piece is face-down, regardless of the color of the piece, as shown in Figure 2.
  - The Cannon *cannot* move (i.e., one square up, down, left, or right), unless it is a capturing action.

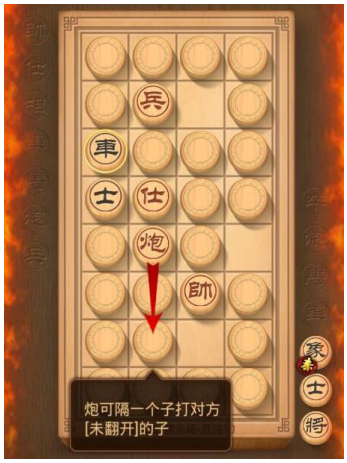

Figure 2: An Example of the Cannon.

4. Each piece has a certain number of points, as shown in Table 1. When a player uses one of her or his pieces to capture an opposing piece, the player gets the points of the captured piece. The player who *gets 60 points first* wins the game.

Project Requirements

You are required to *form a group of two students* (within the same lab class, no exception). Both students need to attend the project defense on Week 15 or 16. Please design and implement a Java program to simulate the Dark chess game for two players. Note that the only programming language you can use in this project is Java.

There are four tasks below to accomplish. A framework/skeleton of the game will be released shortly to facilitate your programming.

Task 1: Game Initialization (10 pts)

- Your program should be able to initialize a new chess game, which includes the chessboard, shuffling the pieces, and randomly placing the pieces face-down on the board.
- Your program should be able to display the status of the game (in progress, red's turn, or black's turn, and their current points etc.)
- Your program should be able to restart a game by clicking a button, rather than closing it and open the game again.

Task 2: Loading and Saving a Game (20 pts)

- Your program should be able to load an existing game from a text file with a pre-defined format by clicking buttons. After loading, all pieces should be placed at their positions given in the text file. The save file includes at least the current chessboard, the previous moves, and the current side to play (red or black).
- Your program should be able to perform error check, e.g., there is no winner yet, any move is invalid, etc.

项目测试用例准备

- Your program should be able to save the current game into a text file.

Task 3: Playing the Game (40 pts)

- Your program should detect the winning status of the game, and end the game when there is a winner.
- Your program should allow pieces to move according to the rules.
- All the pieces that are captured should be displayed beside the board, separated by red and black. For example, three black Soldiers and two black Ministers were captured in Figure 1.
- During one game, your program should be able to switch between the normal mode and the cheating

mode. In the cheating mode, the game allows a player to peek arbitrary face-down pieces on the board.

Task 4: Graphical User Interface (10 pts)

- Your program should have a graphical user interface using Java Swing.
  - JavaFX is acceptable, but it is not allowed to use WebView in JavaFX and implement the GUI with HTML+CSS+JavaScript. You are required to use Java to implement the drawing logic of the interface instead of web programming.

Bonus (20/30 pts)

If your program meets all the above basic requirements, you will get 80 points. The remaining 20/30 points will be given as bonus. You are highly encouraged to go beyond our requirements. Below are some possible ways to get bonus. Compare to the bonus points, the basic points are easier to get. Here you need to rely on your own ability to present your programming charm!

- Design Human vs. AI mode of different difficulty levels, and make the AI player smarter.
- Design a platform for your game, such as adding multi-user, ranking list, adding start menu for selecting the game modes, etc.
- Make your game looks nicer, such as changing the theme, adding sound effect, adding background music, adding more prompt label when the game is in process.
- Show possible moves when a piece is selected.
- Play the process of moving, capturing, and flipping pieces after loading the save file.
- Undo one previous move or multiple previous moves.
- Support online mode in Local Area Network.
- Pack the game as an executable that can be executed on a computer without JRE.
- More...

During project evaluation, your bonus implementations will be evaluated according to their difficulty and novelty, and graded into A/B/C/D class, i.e., 12/8/4/2 points, respectively. Points granted from the same class by multiple bonus functions can stack. All bonus functions count towards the bonus points you can get. The bonus part *is capped at 30 points if your project is demonstrated on the Week 15's labs, and capped at 20 points if it is on the Week 16's labs.* The total grade of project part cannot exceed 100 points.

Notes

1. If you have any questions about the rules of Dark chess, please do ask the instructors. If your program does not match the rules, we may deduct some points from your final project grade.
2. Please start the project as early as possible. Do not leave the project to the last minute.
3. Please form your team as early as possible.

References

1. Banqi Wikipedia. https://en.wikipedia.org/wiki/Banqi.