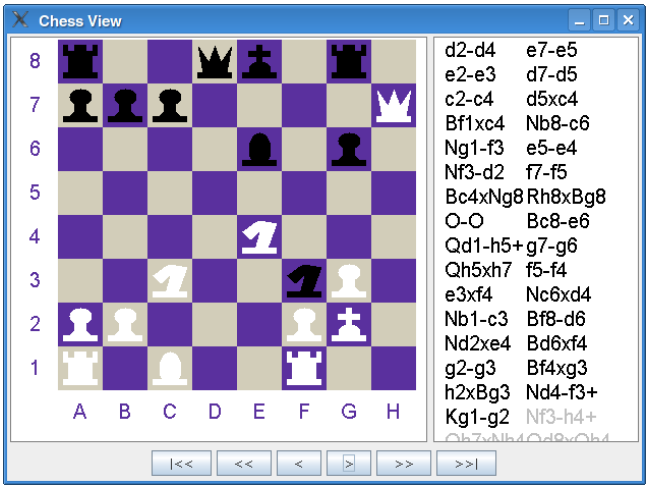
The Chess View System

The `ChessView` system is a simple Java application for viewing chess games written in *long algebraic chess notation*. The following shows a screenshot from `ChessView`:



`ChessView` has only one window, which allows the user to move forward and backward through a chess game. The moves of the game, written in long algebraic chess notation, are given in the rightmost pane of the window; the current state of the chess board is shown in the leftmost pane.

Long Algebraic Notation

Long algebraic chess notation is a way for writing down the moves taken during a chess game. The following illustrates the start of a game in this notation:

White	Black
`e2-e4`	`e7-e5`
`d2-d4`	`e5xd4`
`Nb1-c3`	`Qd8-f6`

The first move by White, `e2-e4`, indicates the pawn at position `e2` will advance to position `e4`. When indicating pawn movement, no piece specifier is given. However, when indicating the movement of

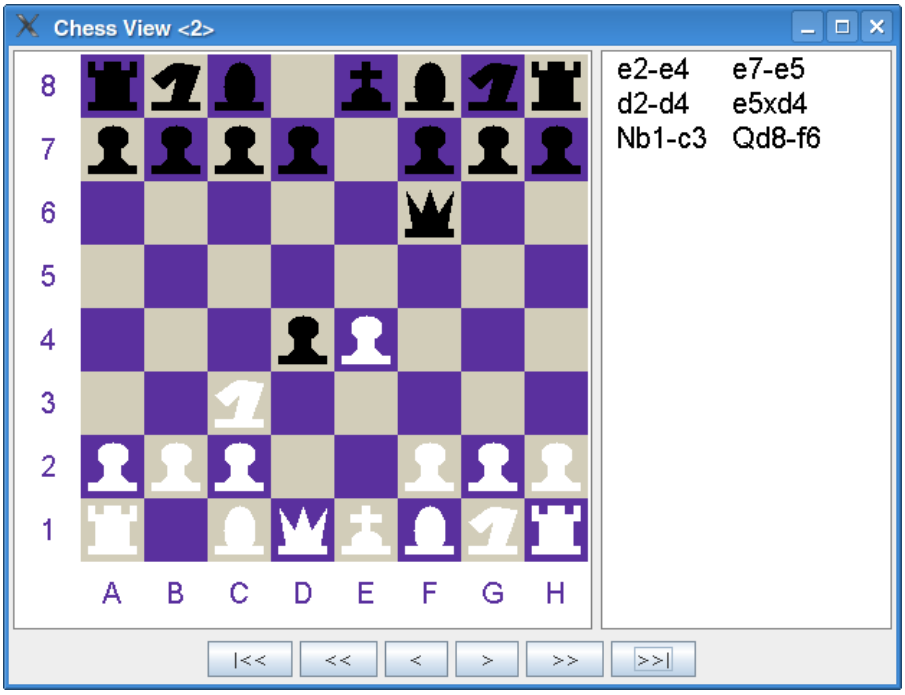


Figure 1: Illustrating the state of a game after 3 rounds.

other pieces a specifier is always given. The specifiers are: `N`=kNight; `B`=Bishop; `R`=Rook; `Q`=Queen; `K`=King. So, for example, White's last move above, `Nb1-c3`, indicates his Knight moves from `b1` to `c3`. Finally, a move where one piece takes another is indicated using a `x`, as in Black's move `e5xd4` — the (black) pawn at position `e5` takes the (white) pawn at `d4`. Figure 1 shows the state of the board after the moves in the above game have been made.

There are a number of other moves which can be made during a game of chess, including: *putting a king in check* (e.g `Qe1-h4+`), *castling* (e.g. `O-O`), *pawn promotion* (e.g. `b7-b8=N`) and *en passant* (e.g. `b4xa3ep`). If you are not familiar with the rules of Chess, the following links provide an excellent starting point:

```
http://en.wikipedia.org/wiki/Chess
http://en.wikipedia.org/wiki/Chess_notation
http://en.wikipedia.org/wiki/Algebraic_chess_notation
```

You can also find many other good resources on the Internet regarding the game of chess.

Some notes on these packages:

- The `assignment3/chessview/` package contains the high-level game classes, including those for representing the chess board and the game itself.
- The `assignment3/chessview/pieces/` packages contains a class for each of the different chess pieces. These contain code related to the movement of the pieces.
- The `assignment3/chessview/moves/` packages contains a class for each of the different kind of move that can be made in the game. These contain code related to structuring a move, and ensuring it is valid.
- The `assignment3/chessview/viewer/` package contains code related to the ChessView interface.

The class `assignment3/chessview/Board` is one of the main classes in the ChessView system. This is responsible for representing the state of the chess board, including the position of all of the pieces. This uses a 2-dimensional array to represent the chess board, where each cell in that array represents a location on the board. The board also provides methods for determining whether a particular diagonal, horizontal or vertical move will be unobstructed. Figure 2 provides a visualisation of a simple `Board` object.

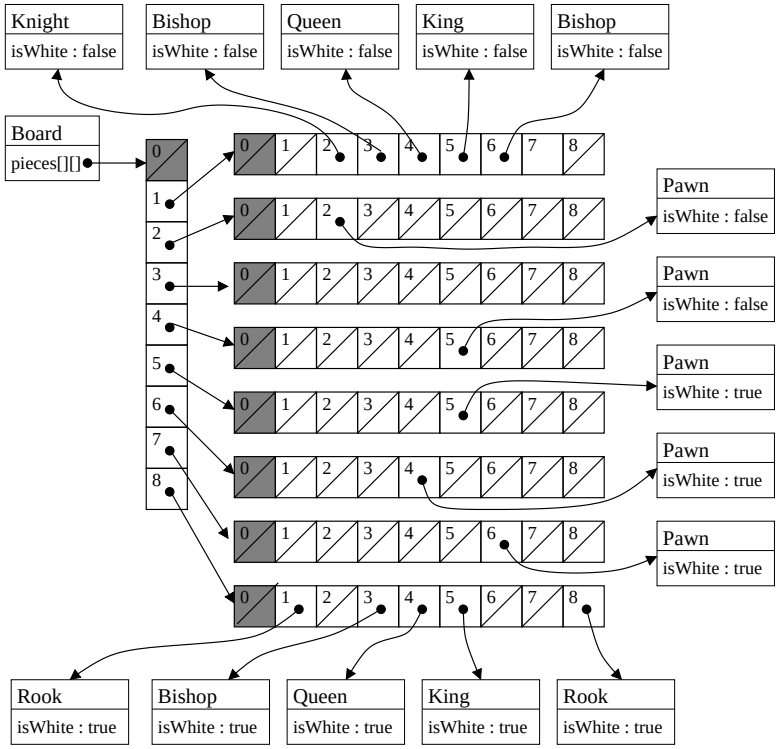


Figure 2: Illustrating an example `Board` object, which contains a 2-dimensional array of pieces. An empty square is indicated by a box with a slash, and this corresponds to the array cell holding `null`. The first element of the row array, and the first element of each column array are unused (marked in gray). This allows the row numbers to line up with those of the normal chess board. Column `a` of the chess board corresponds to index 1 in the column array, and so on for columns `b`, `c`, . . . , etc.