

Console Chess

Design a chess game that can be played on the console.

The chessboard should be represented by a class **ChessBoard**. For the various chess pieces own classes **King**, **Queen**, ... are to be developed, which are derived from the abstract base class **ChessPiece**.

Translations:

- King: König
- Queen: Dame
- Bishop: Läufer
- Knight: Springer
- Rook: Turm
- Pawn: Bauer

The method **CanMove(int xTo, int yTo)** returns whether the chess piece can or cannot make the move.

The moves should be entered in the form "**e2-e4**".

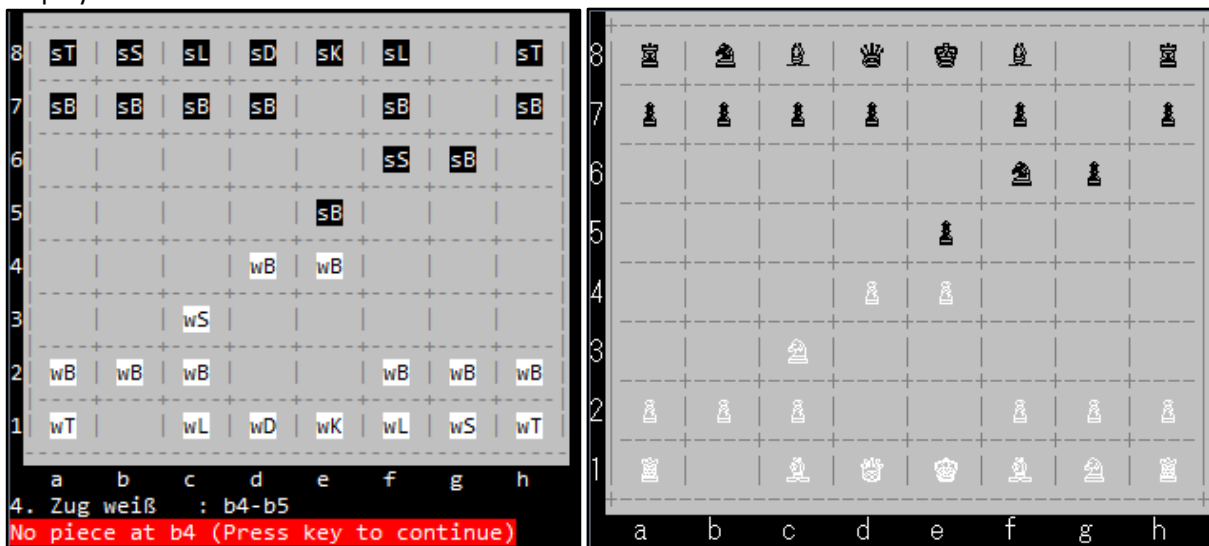
For saving the figure positions you can choose one of two variants:

- in the **ChessBoard** class all chess pieces in a 2d array **ChessPiece[,]**
- store the coordinates per chess piece

The following should be considered:

- Only chess pieces that are of the correct color may be drawn (i.e. only a white chess piece when white is on)
- **no own chess pieces** may be **hit**
- For individual chess pieces, it should be checked whether the specified **target field can be reached** by the corresponding chess piece. At least for Knight and King this must be tested.
- You can draw a chess piece either by its "name" or by a corresponding symbol (unicode see https://en.wikipedia.org/wiki/Chess_symbols_in_Unicode). In that case the Encoding of the console has to be set to UTF8: `Console.OutputEncoding = System.Text.Encoding.UTF8`

Display:



Please pay attention to the following when displaying the board:

- White chess pieces with a white background, black ones with a black background (`Console.ForegroundColor`, `Console.BackgroundColor`)
- Background of the chessboard should be gray
- You can access any position on the screen with `Console.SetCursorPosition`
- Error messages are to be displayed in red