

```

1 package chess.pieces;
2
3 import boardgame.Board;
4 import boardgame.Piece;
5 import boardgame.Position;
6 import chess.ChessPiece;
7 import chess.Color;
8
9 import java.util.Scanner;
10
11 public class Rook extends ChessPiece {
12
13     public Rook(Board board, Color color) {
14         super(board, color);
15     }
16
17     @Override
18     public String toString() {
19         return "R";
20     }
21
22     public boolean[][] possibleMoves(Position position){
23         boolean[][] mat = new boolean[getBoard().getRows()][getBoard().getColumns()];
24         Position p = new Position(0,0);
25
26         // above
27         p.setValues(position.getRow()-1, position.getColumn());
28         while (p.getRow() >= 0 && !getBoard().thereIsAPiece(p) ){
29             mat[p.getRow()][p.getColumn()] = true;
30             p.setRow(p.getRow()-1);
31         }
32         if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
33             mat[p.getRow()][p.getColumn()] = true;
34         }
35
36         // left
37         p.setValues(position.getRow(), position.getColumn()-1);
38         while (p.getColumn() >= 0 && !getBoard().thereIsAPiece(p) ){
39             mat[p.getRow()][p.getColumn()] = true;
40             p.setColumn(p.getColumn()-1);
41         }
42         if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
43             mat[p.getRow()][p.getColumn()] = true;
44         }
45
46         // below
47         p.setValues(position.getRow()+1, position.getColumn());
48         while (p.getRow() < getBoard().getRows() && !getBoard().thereIsAPiece(p) ){
49             mat[p.getRow()][p.getColumn()] = true;
50             p.setRow(p.getRow()+1);
51         }
52         if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
53             mat[p.getRow()][p.getColumn()] = true;
54         }
55
56         // right
57         p.setValues(position.getRow(), position.getColumn()+1);
58         while (p.getColumn() < getBoard().getColumns() && !getBoard().thereIsAPiece(p) ){
59             mat[p.getRow()][p.getColumn()] = true;
60             p.setColumn(p.getColumn()+1);
61         }
62         if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
63             mat[p.getRow()][p.getColumn()] = true;
64         }
65
66         return mat;
67     }
68
69     @Override
70     public boolean[][] possibleMoves() {
71         return possibleMoves(position);
72     }
73 }
74
75 }
76

```