

File - C:\aprendizado\java\nélioUdemy\chess-system\src\chess\pieces\Bishop.java

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