```java
package chess.pieces;

import boardgame.Board;
import boardgame.Position;
import chess.ChessPiece;
import chess.Color;

public class Bishop extends ChessPiece {
    public Bishop(Board board, Color color) {
        super(board, color);
    }
    @Override
    public boolean[][] possibleMoves() {
        return possibleMoves(position);
    }

    public boolean[][] possibleMoves(Position position) {
        boolean[][] mat = new boolean[getBoard().getRows()][getBoard().getColumns()];
        Position p = new Position(0,0);

        // left-up
        p.setValues(position.getRow()-1, position.getColumn()-1);
        while (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setValues(p.getRow()-1, p.getColumn()-1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // left-down
        p.setValues(position.getRow()+1, position.getColumn() -1);
        while (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setValues(p.getRow()+1, p.getColumn()-1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // right-up
        p.setValues(position.getRow()-1, position.getColumn()+1);
        while (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setValues(p.getRow()-1, p.getColumn()+1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        // right-down
        p.setValues(position.getRow()+1, position.getColumn() +1);
        while (getBoard().positionExists(p) && !getBoard().thereIsAPiece(p) ){
            mat[p.getRow()][p.getColumn()] = true;
            p.setValues (p.getRow()+1, p.getColumn()+1);
        }
        if (getBoard().positionExists(p) && isThereOpponentPiece(p)){
            mat[p.getRow()][p.getColumn()] = true;
        }

        return mat;
    }

    @Override
    public String toString() {
        return "B";
    }
}

```