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

```java
package chess.pieces;

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

public class Knight extends ChessPiece {
    public Knight(Board board, Color color) {
        super(board, color);
    }

    private boolean canMove(Position position){
        ChessPiece p = (ChessPiece) getBoard().piece(position);
        return p == null || p.getColor() != getColor();
    }

    @Override
    public boolean[][] possibleMoves() {
        boolean[][] mat = new boolean[getBoard().getRows()][getBoard().getColumns()];

        // 2-up-left
        Position p = new Position(position.getRow()-2, position.getColumn()-1);
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}
        p.setColumn(p.getColumn()+2); // 2-up-rigth
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}

        // 2-down-left
        p.setValues(position.getRow()+2, position.getColumn()-1);
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}
        p.setColumn(p.getColumn()+2); // 2-up-rigth
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}

        // 2-left-up
        p.setValues(position.getRow()-1, position.getColumn()-2);
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}
        p.setRow(p.getRow()+2); // 2-left-down
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}

        // 2-right-up
        p.setValues(position.getRow()-1, position.getColumn()+2);
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}
        p.setRow(p.getRow()+2); // 2-right-down
        if (getBoard().positionExists(p)) {mat[p.getRow()][p.getColumn()] = canMove(p);}

        return mat;
    }

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

```