Testing Log for Chess Project

The purpose of this document is to keep track of all the testing (manual or via a test suite) that you’ve performed on your chess game to ensure it works correctly. Use the below grid as you go to record each thing you tested. You should be sure to test all functions as fully as possible!

Manual Tests:

| Name of function you’re testing | Action taken | Expected Response | Pass/Fail | Screen shot |
| --- | --- | --- | --- | --- |
| Board constructor | Press “start” button | A board is displayed on the screen with a white square in the top left corner and a checkerboard pattern across the rest of the 8 rows/columns. | Pass |  |
| initializePieces() | Press “start” button | 8 black pawns are placed on the first row, and 8 white pawns are placed on the last row. | Pass |  |
| getLegalMoves() | Dragging a piece  (A for loop including print is at the end of the method )  for(Square sq : arr) {  System.out.println(sq.getYNum() + " / " + sq.getXNum());  } | the legal positions should be printed | Pass |  |
| mouseReleased() | Dragging a piece | The piece can properly move and capture pieces if its legal  and they should snap back if the move is illegal | Pass |  |
| getControlledSquares() | Dragging a piece  (A for loop including print is in the mouseReleased method and it prints each XPosition and YPosition element of the array ) | The program should print diagonal squares (1 up right and 1 up left)  (based on the fromMoveSquare (previous position)) | Pass |  |
|  |  |  |  |  |
|  |  |  |  |  |

Code tests (optional):

If you made any JUnit Tests please copy and paste them here: