Huq2FinalProject Manual Test Plan version 3

Note: in order for these tests to be particular to the GUI, the models used by the GUI must be functionally correct.

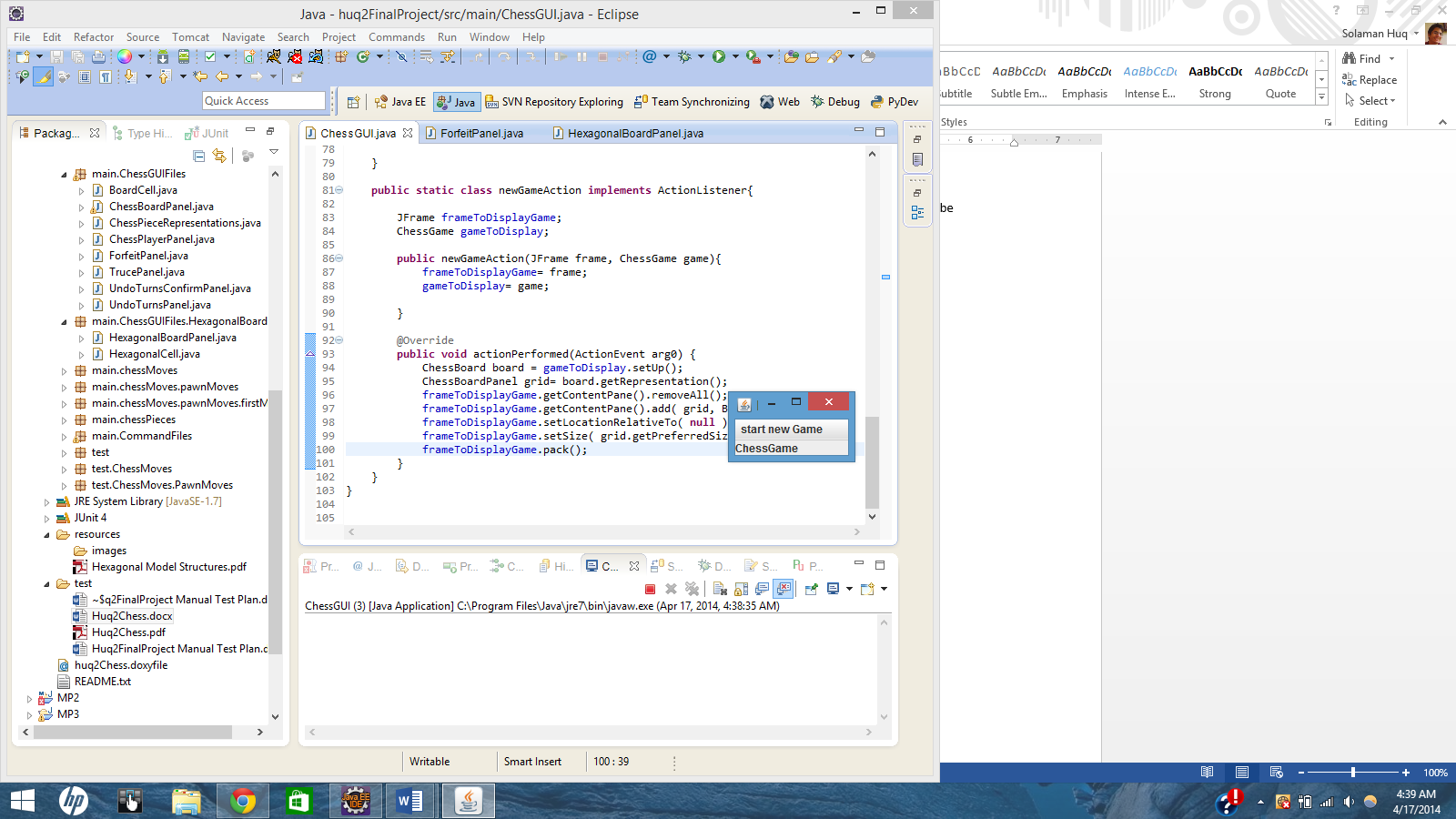
Table of Contents

1. [Standard Gameplay Features](#StandardGamplayFeatures):
   1. [Game Selection](#GameSelection)
   2. [Performing moves](#PerformingMoves)
2. [Rectangular Chess](#RectangularChess)
   1. Standard Chess
3. Hexagonal Chess
   1. Glinski’s Chess
   2. Shafran’s Chess
   3. Performing moves
4. Custom Games:
   1. Set board
   2. Laying Pieces
   3. Giving a name
   4. Load Outs and selecting the game

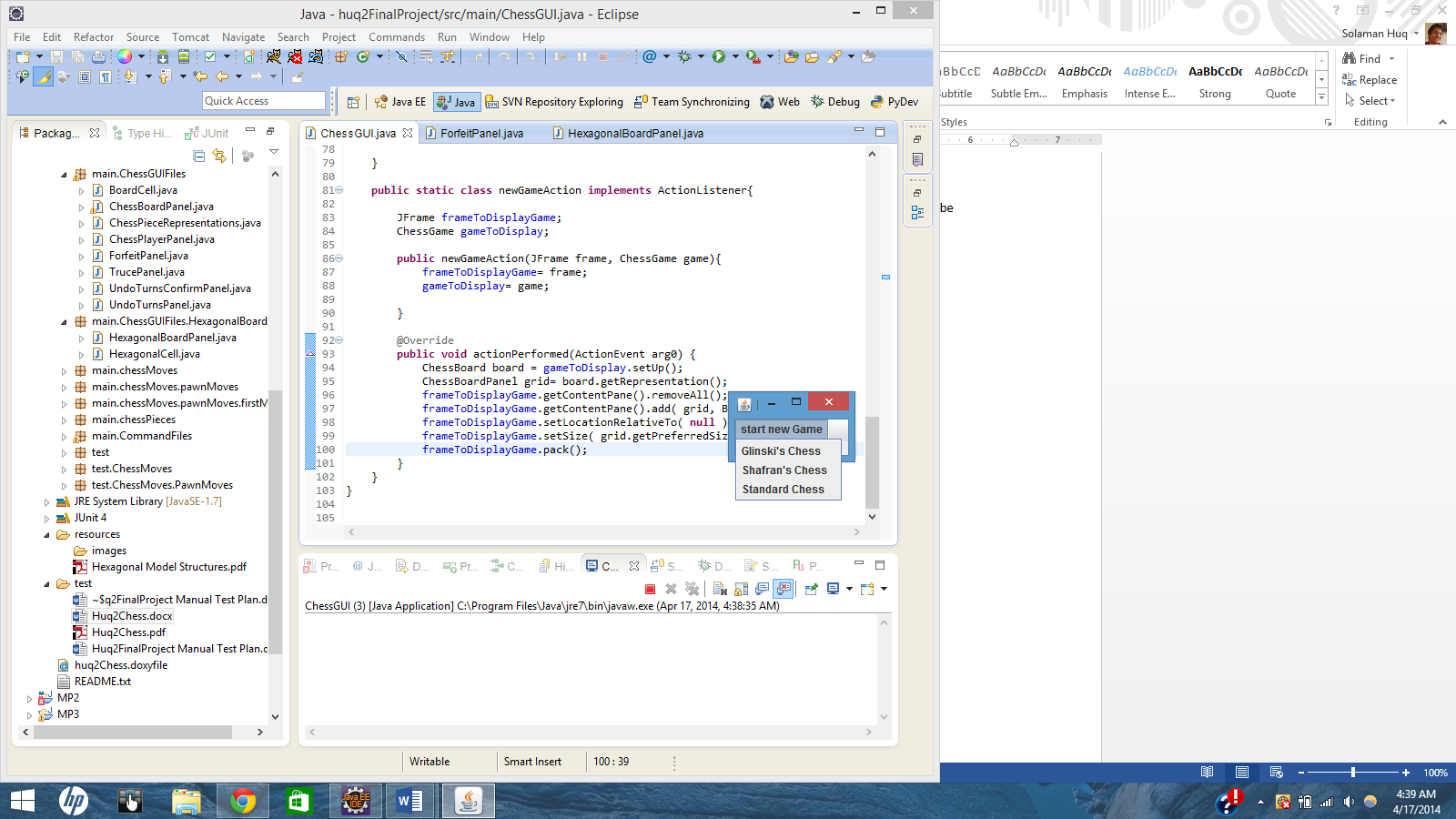
Standard Gameplay Features

1. Game Selection

Starting the application should produce a small menu like so



Click on “start new Game”, this should be in your display. If other games exist, that is fine.

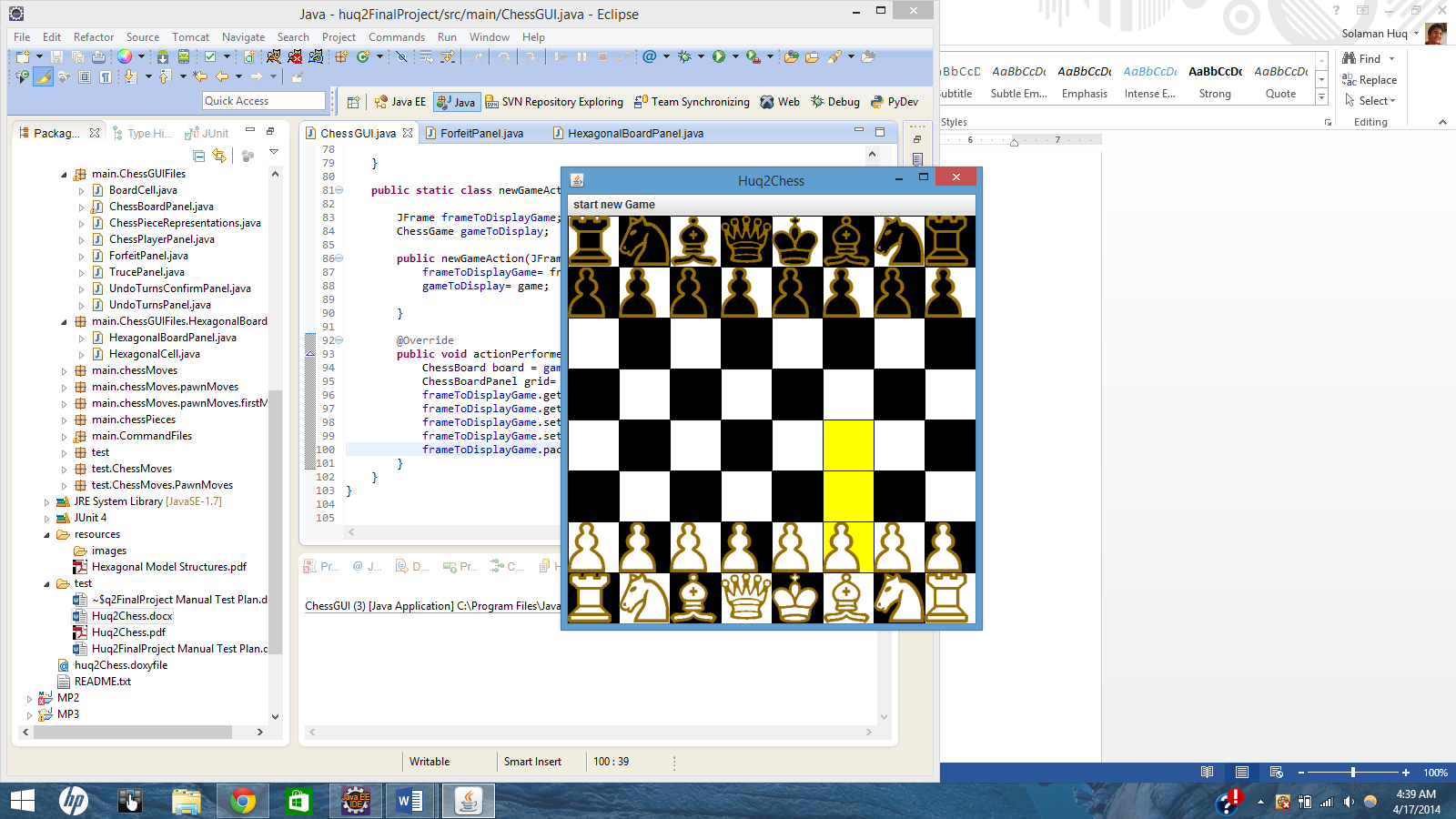


Select a game, that game should now be displayed. Go back to the menu and select a different game. The display should Change to show the new game. This completes the Game Selection Tests.

1. Performing Moves
   1. Highlighting Moves

NOTE: Tests a-c should be repeated for Shafran’s Chess to assure functionality of hexagonal boards. Test d need not be repeated as passing test d in Standard Chess assures that it works for all variants of boards.

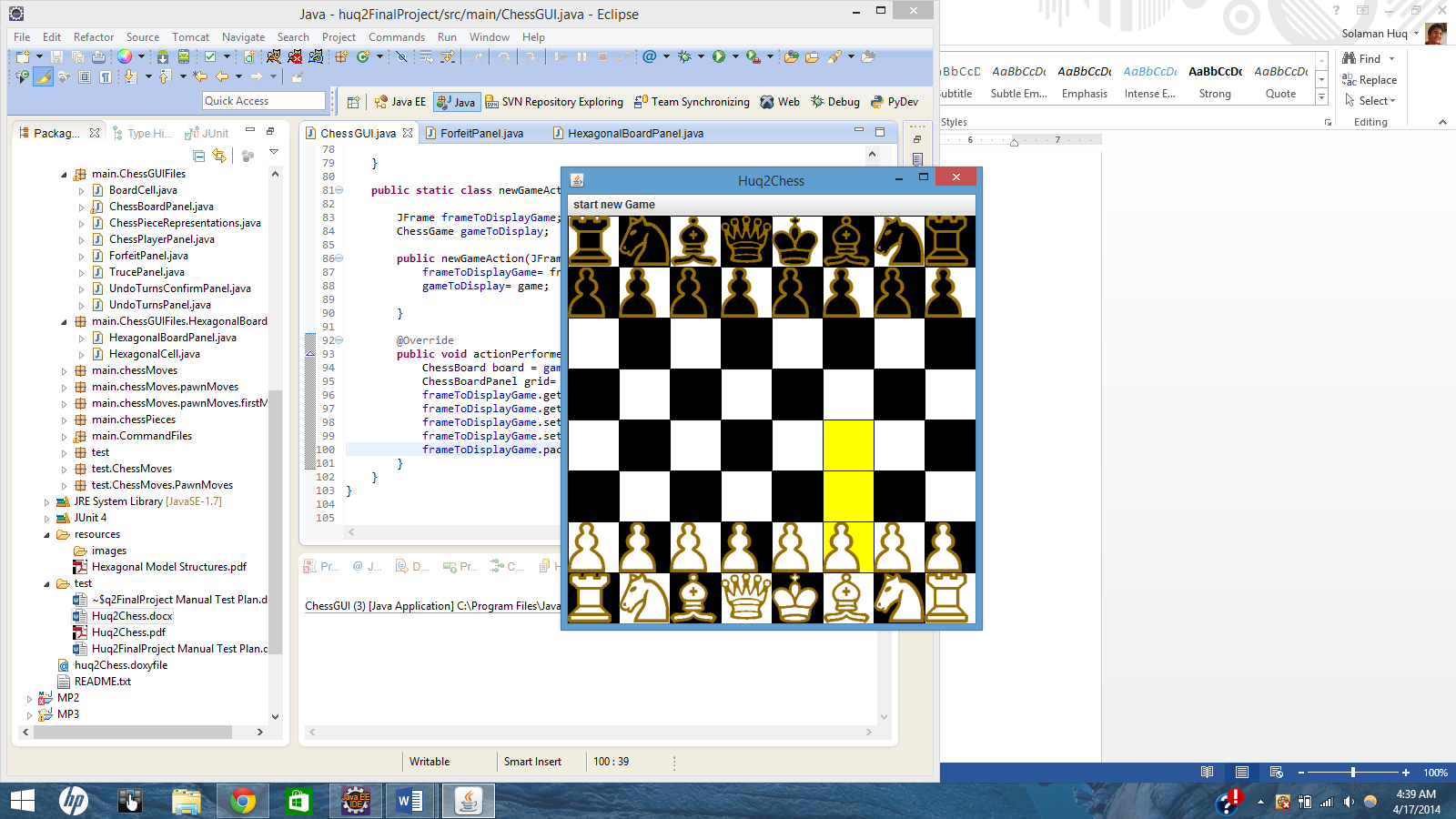
From the Game menu, select Standard Chess. The game displayed should be for white’s turn. Run the mouse over any of the pieces; their available moves should be highlight like so



Running over to another piece should set the spaces back to normal, and should highlight another piece’s moves.

* 1. **Selecting Pieces To Move**

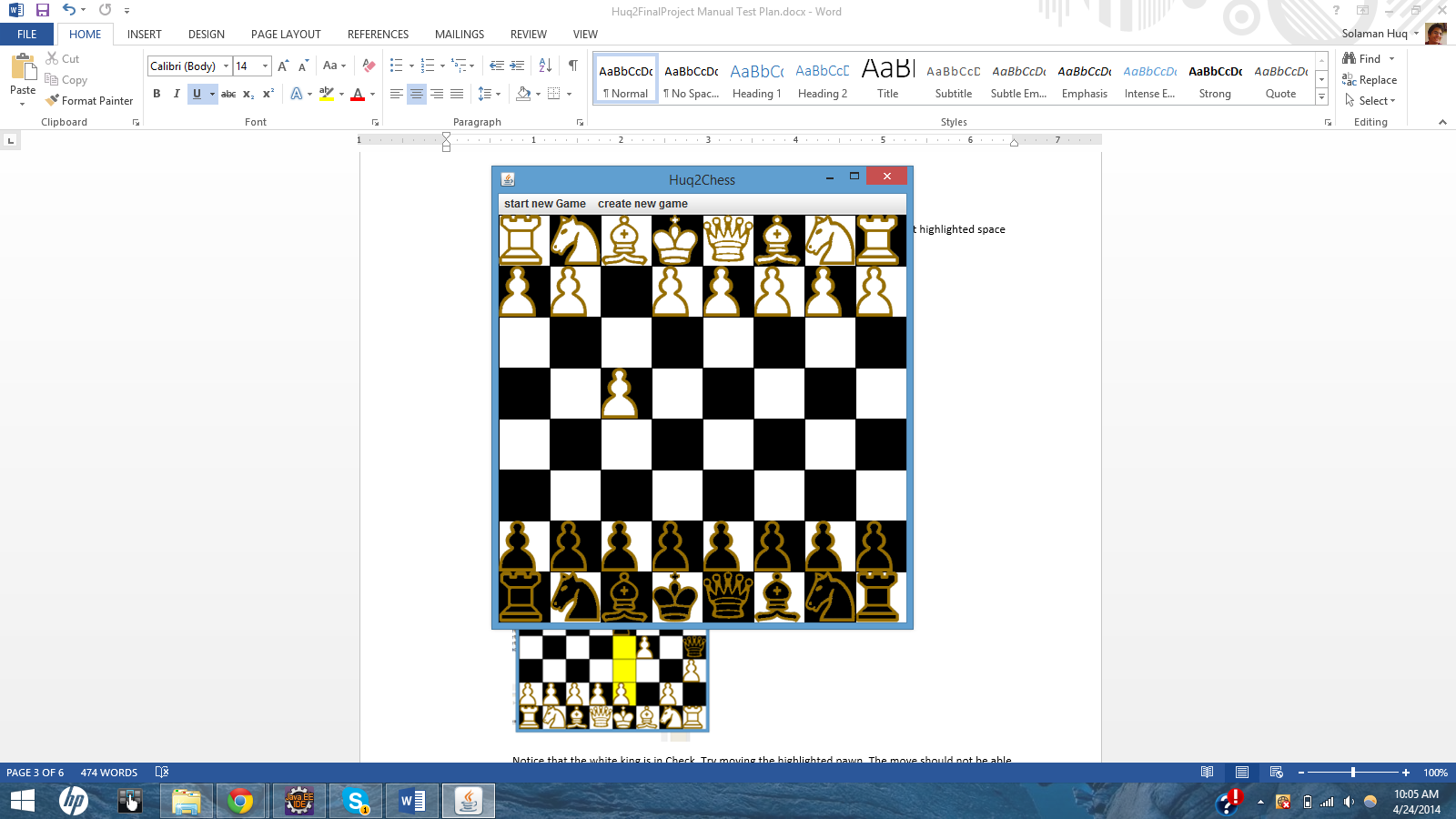
Select the shown Pawn. As you move your mouse, the highlighted selection should remain constant



Click on the piece again, this should return the spaces to their normal color. You can select another piece from here as well.

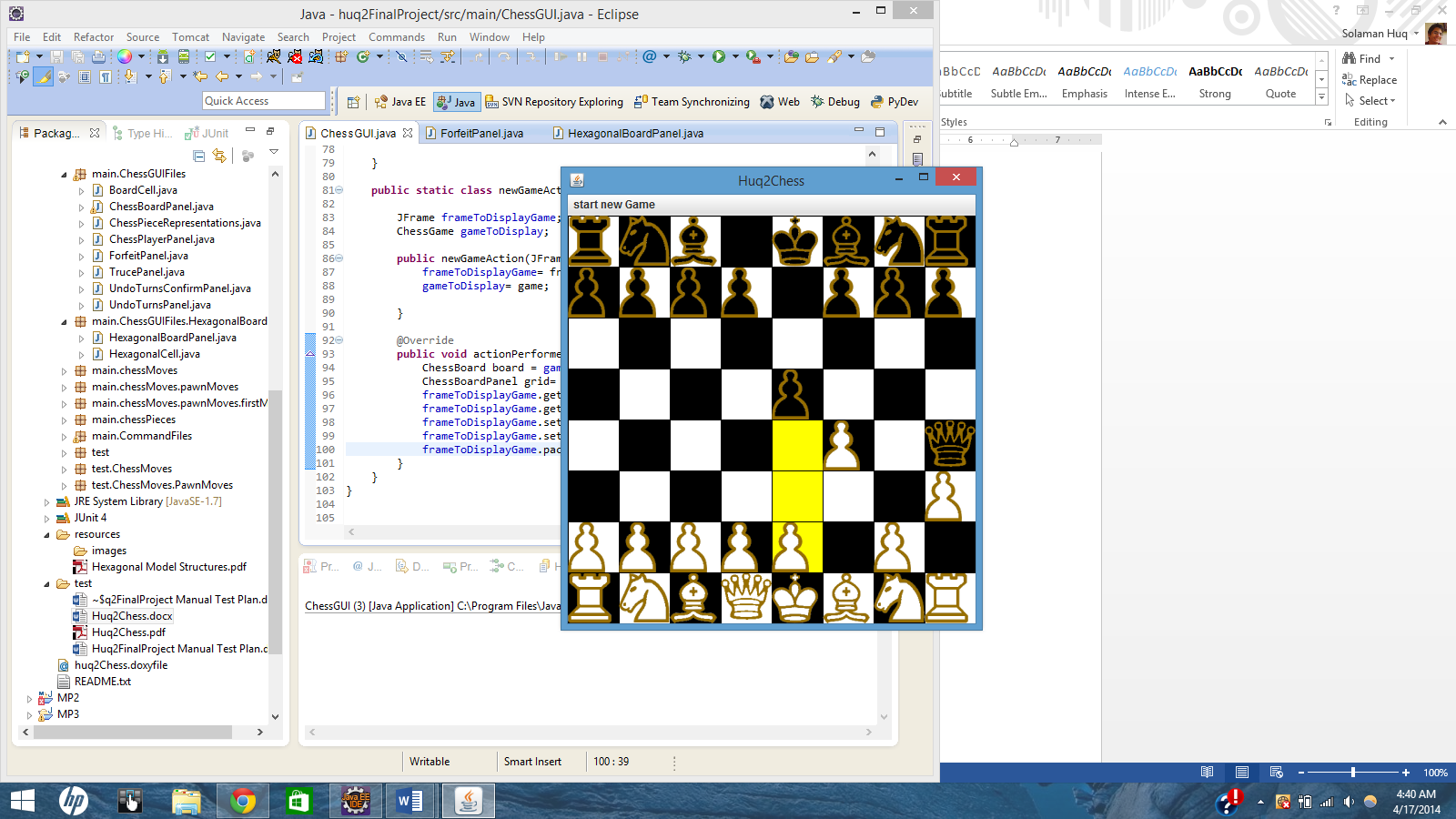
* 1. **Performing the Move**

Select the pawn that was shown from the previous screen, clicked on the topmost highlighted space to move it like so. Note that the display is now reversed for the opponents turn. Check to see that the opponents move’s can be highlighted, and the board is accurately reversed.



* 1. **Performing Moves in Check**

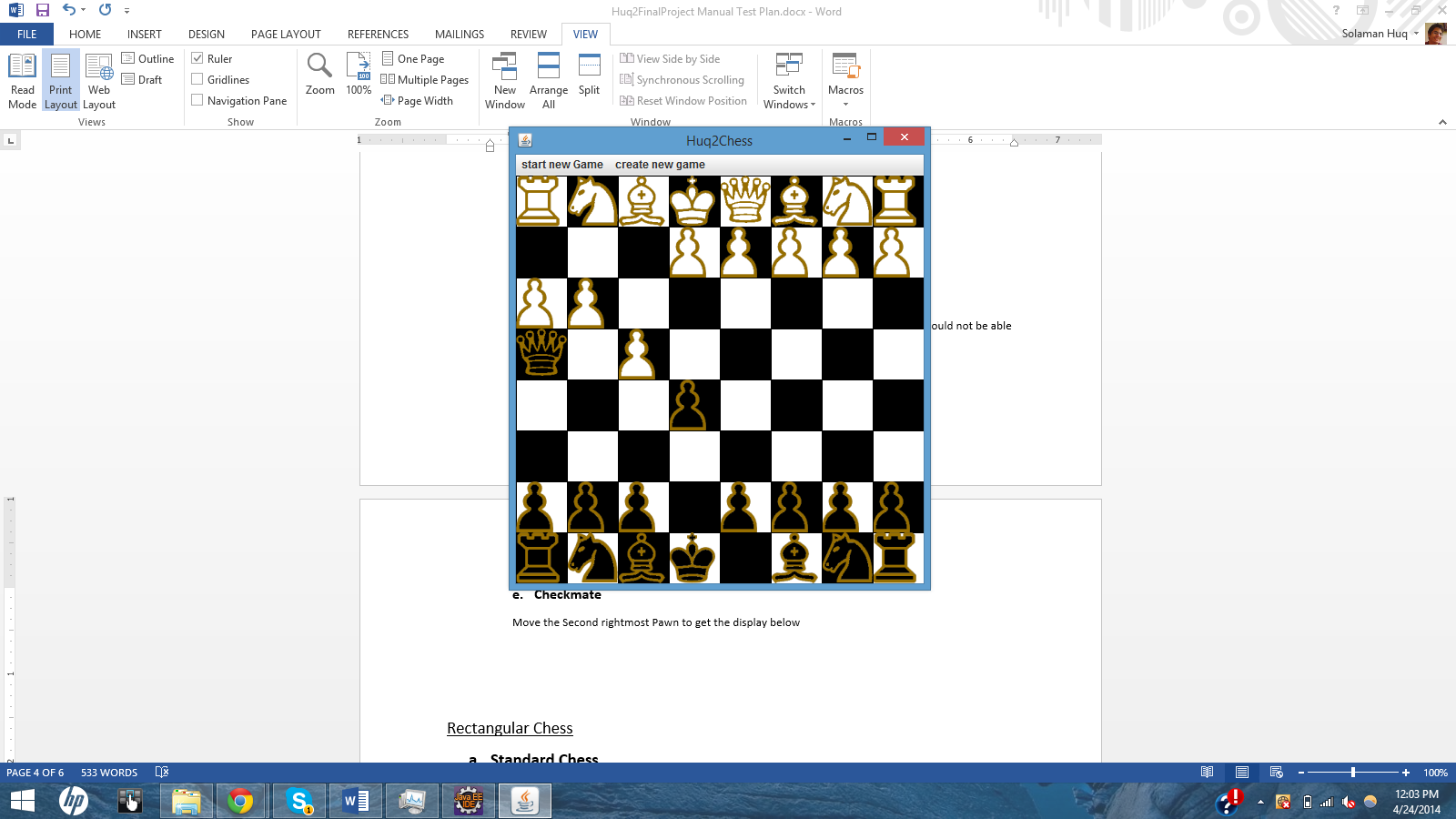
From here, try moving the pieces so that they match the board below



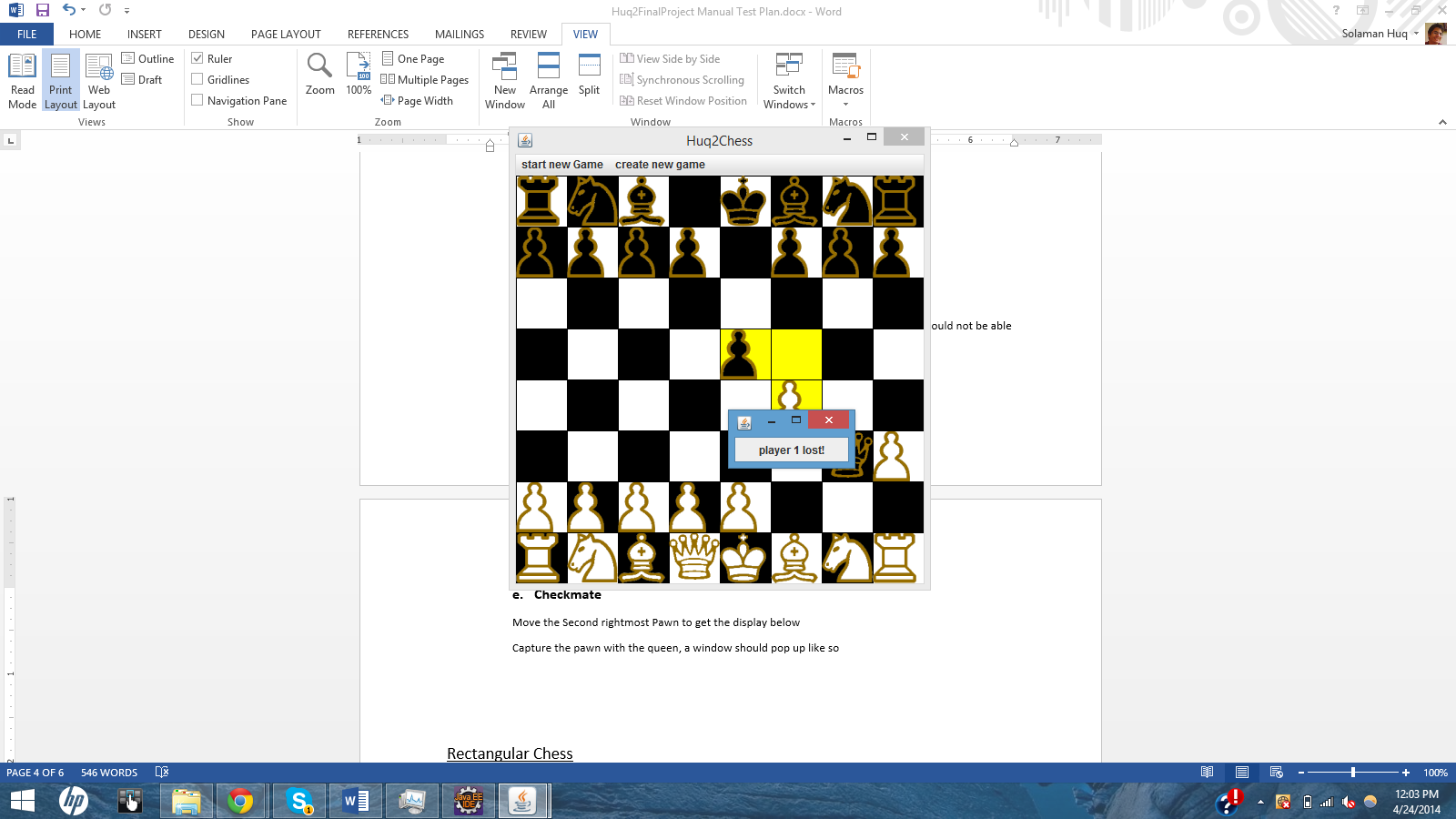
Notice that the white king is in Check. Try moving the highlighted pawn. The move should not be able to be made, and the Pawn should remain where it is.

* 1. **Checkmate**

Move the Second rightmost Pawn to get the display below



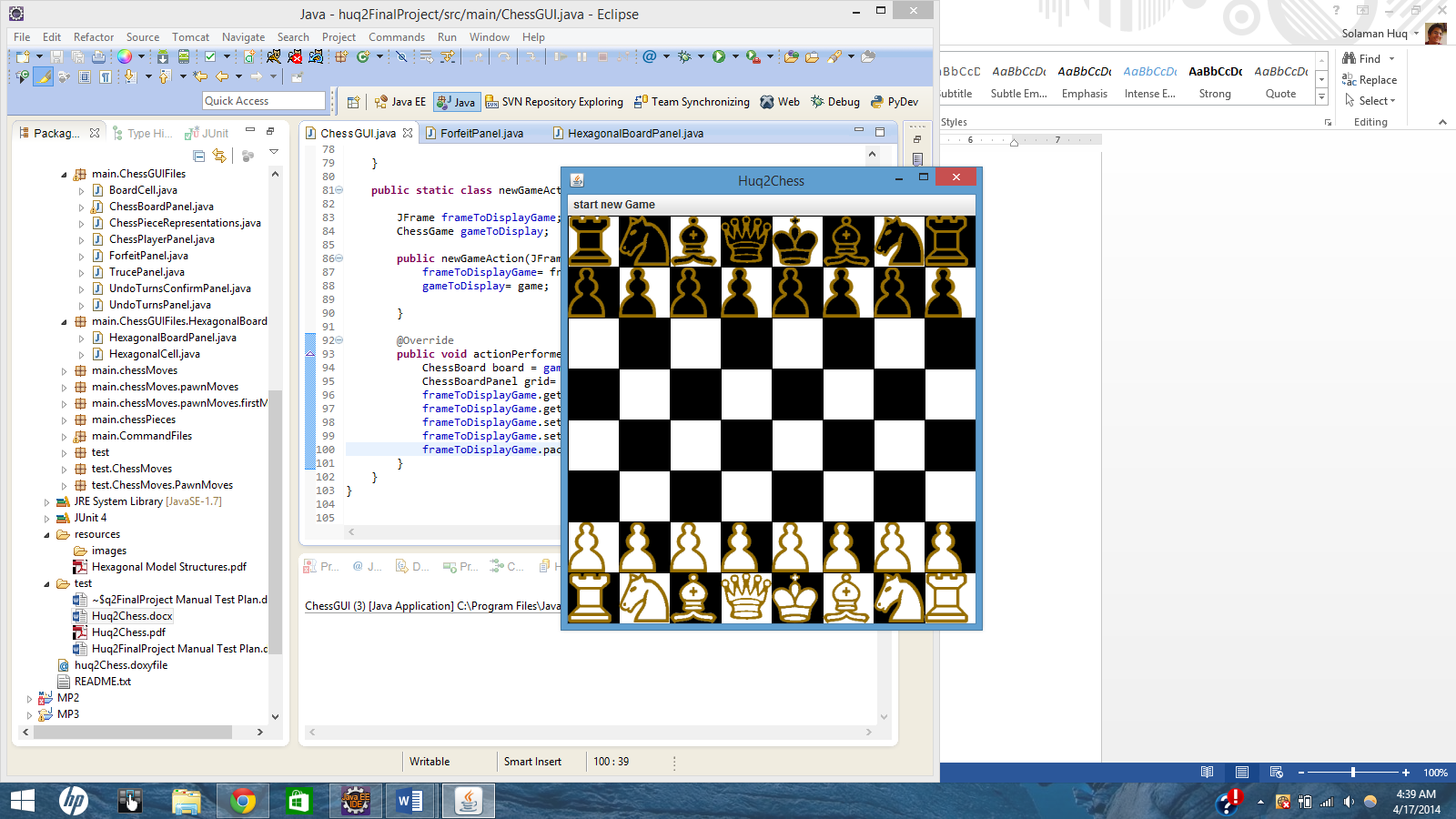
Capture the pawn with the queen, a window should pop up like so



Rectangular Chess

1. **Standard Chess**
   1. **Layout**

While not explicitely stated, Standard Chess should match the layout previously seen, or as such the one below

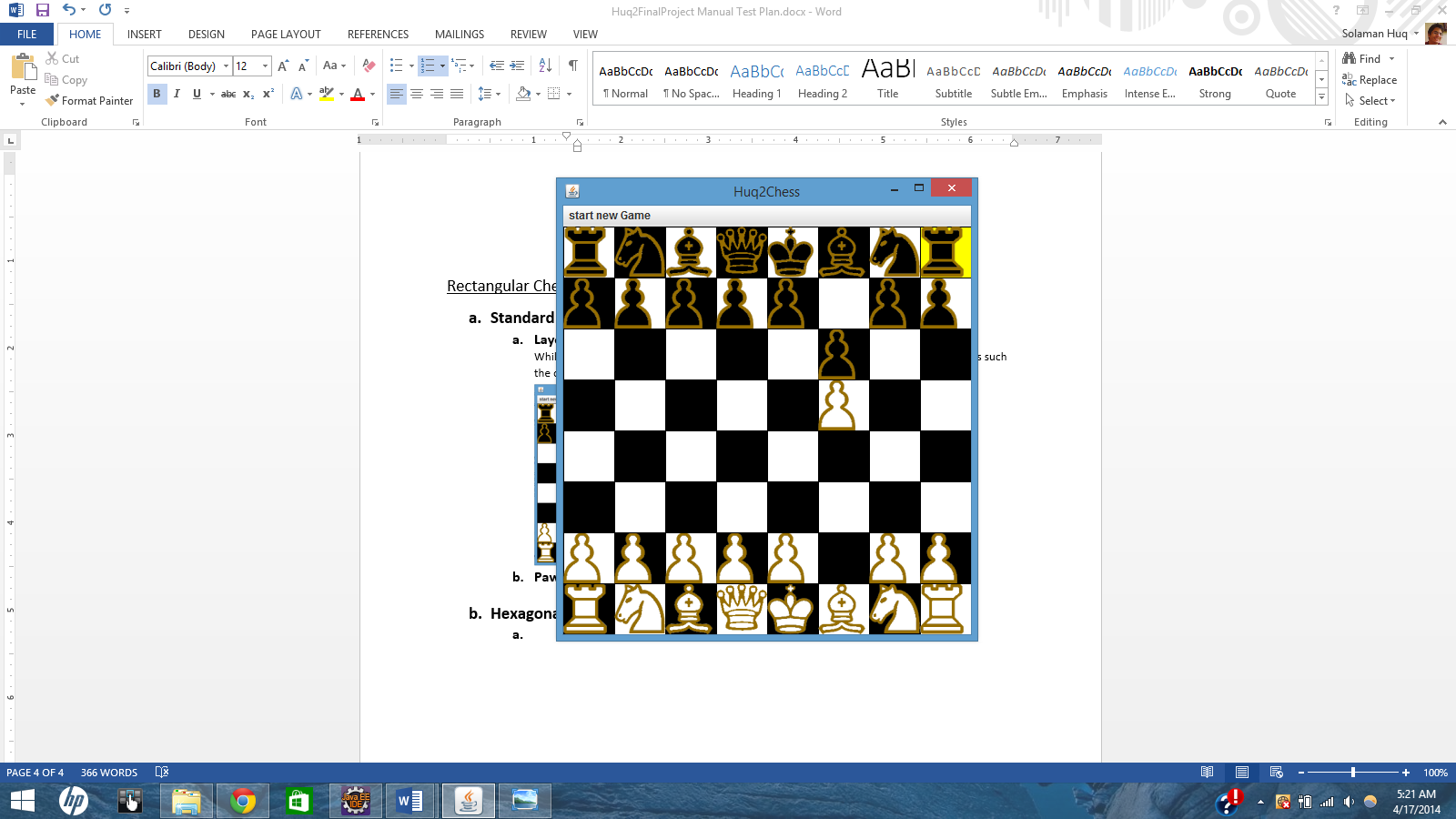


* 1. **Pawn En Passant**

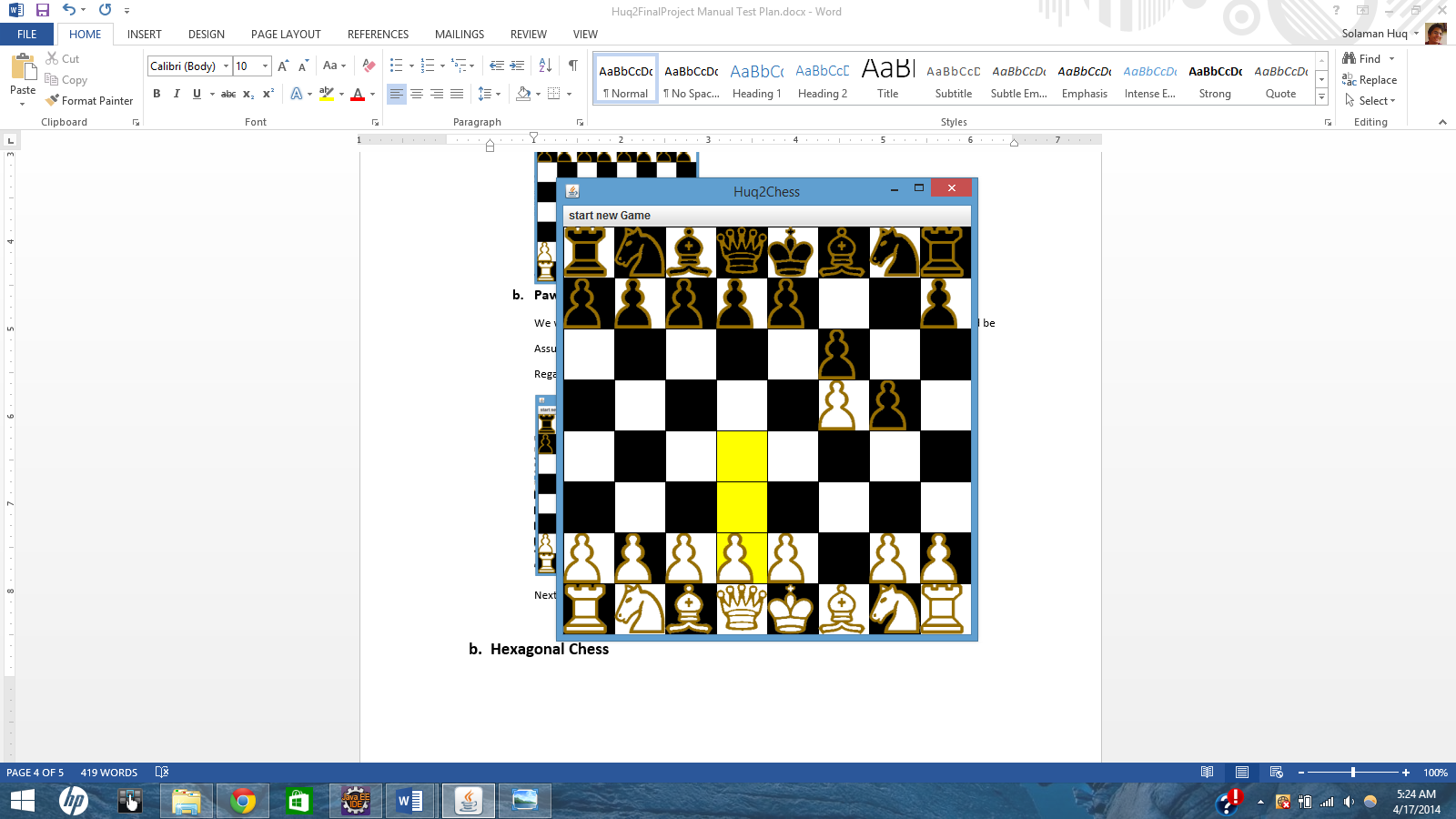
We will not test the functionality of the move, only that it is available. Functionality should be

Assured by model tests.

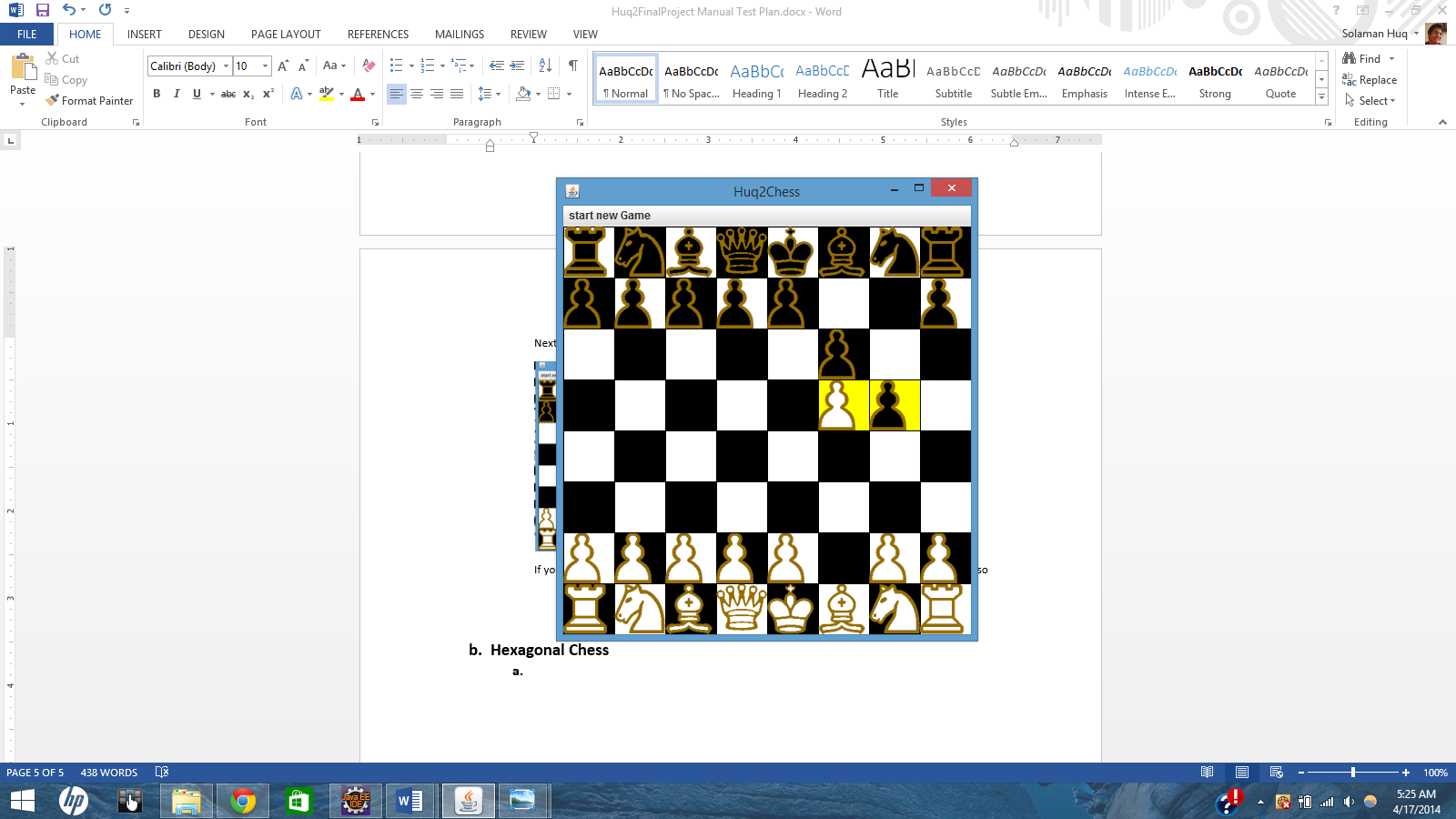
Regardless, with the started game, move the pieces so that they match the display below



Next move the black pawn at (7,7) so that it is next to the white pawn like so



If you click on the white pawn, it should display a highlight for the black pawn’s space like so

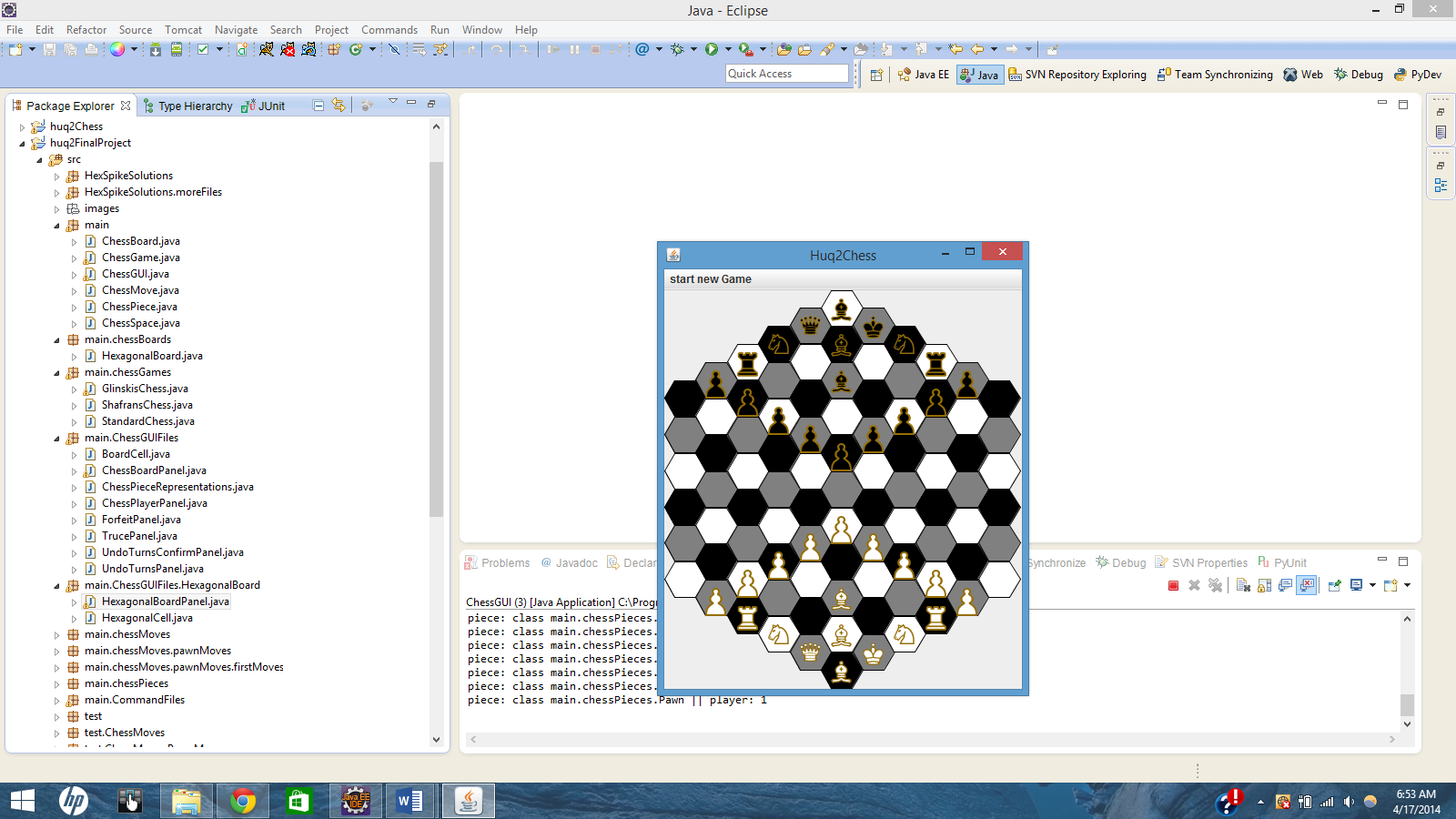


Click on the space, En Passant should be performed.

**Hexagonal Chess**

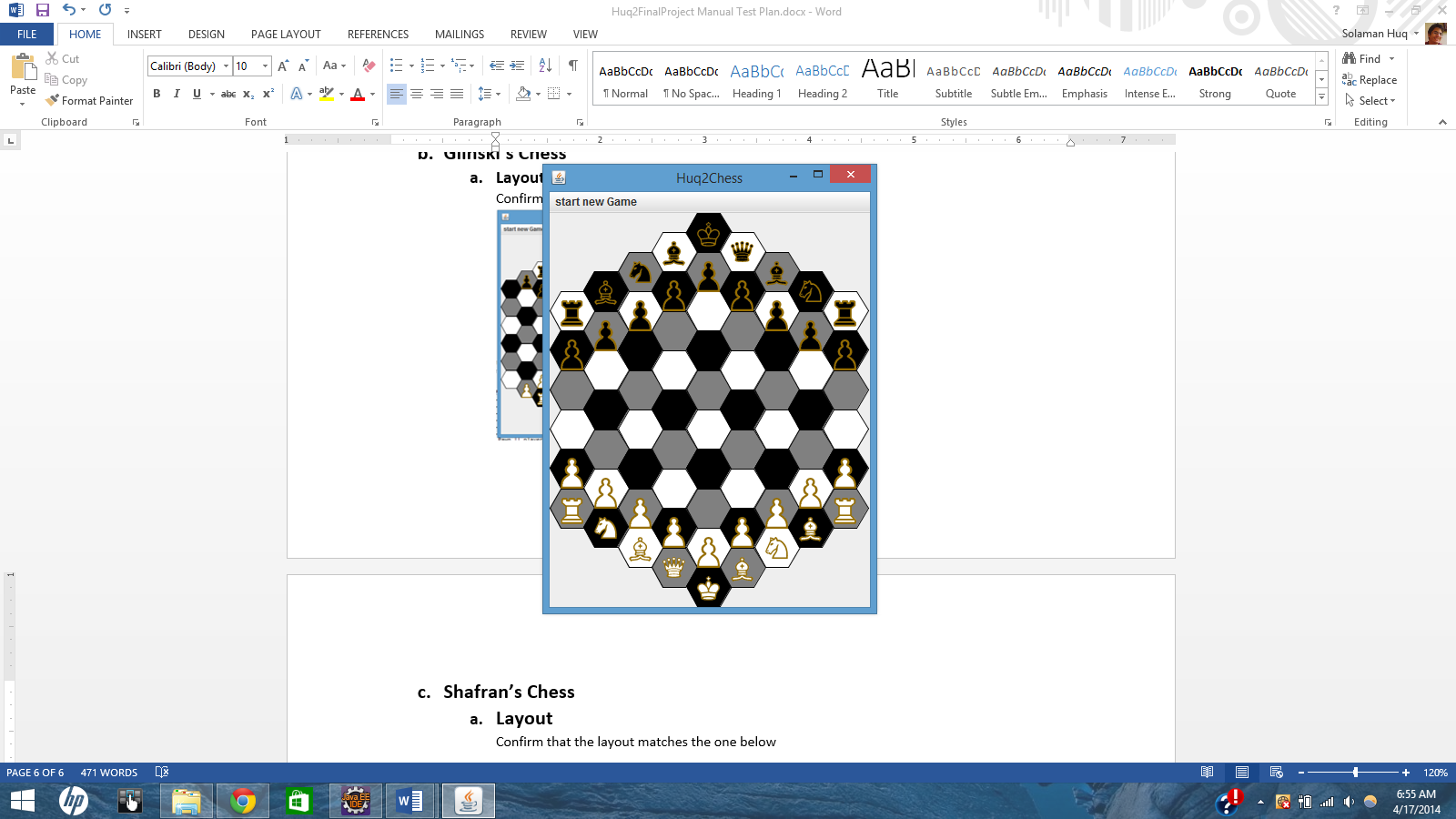
1. **Glinski’s Chess**
   1. **Layout**

Confirm that the layout matches the one below



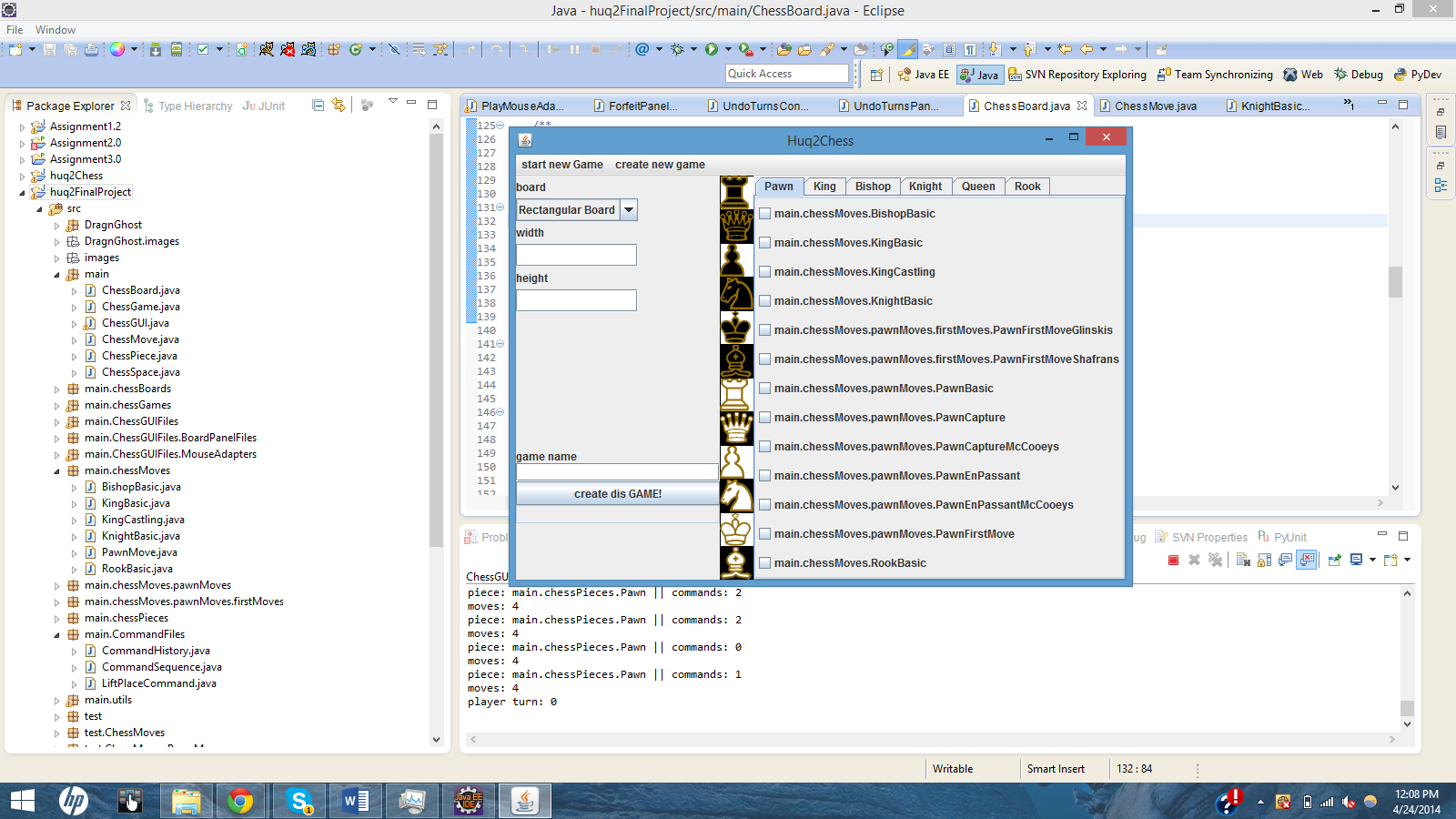
1. **Shafran’s Chess**
   1. **Layout**

Confirm that the layout matches the one below



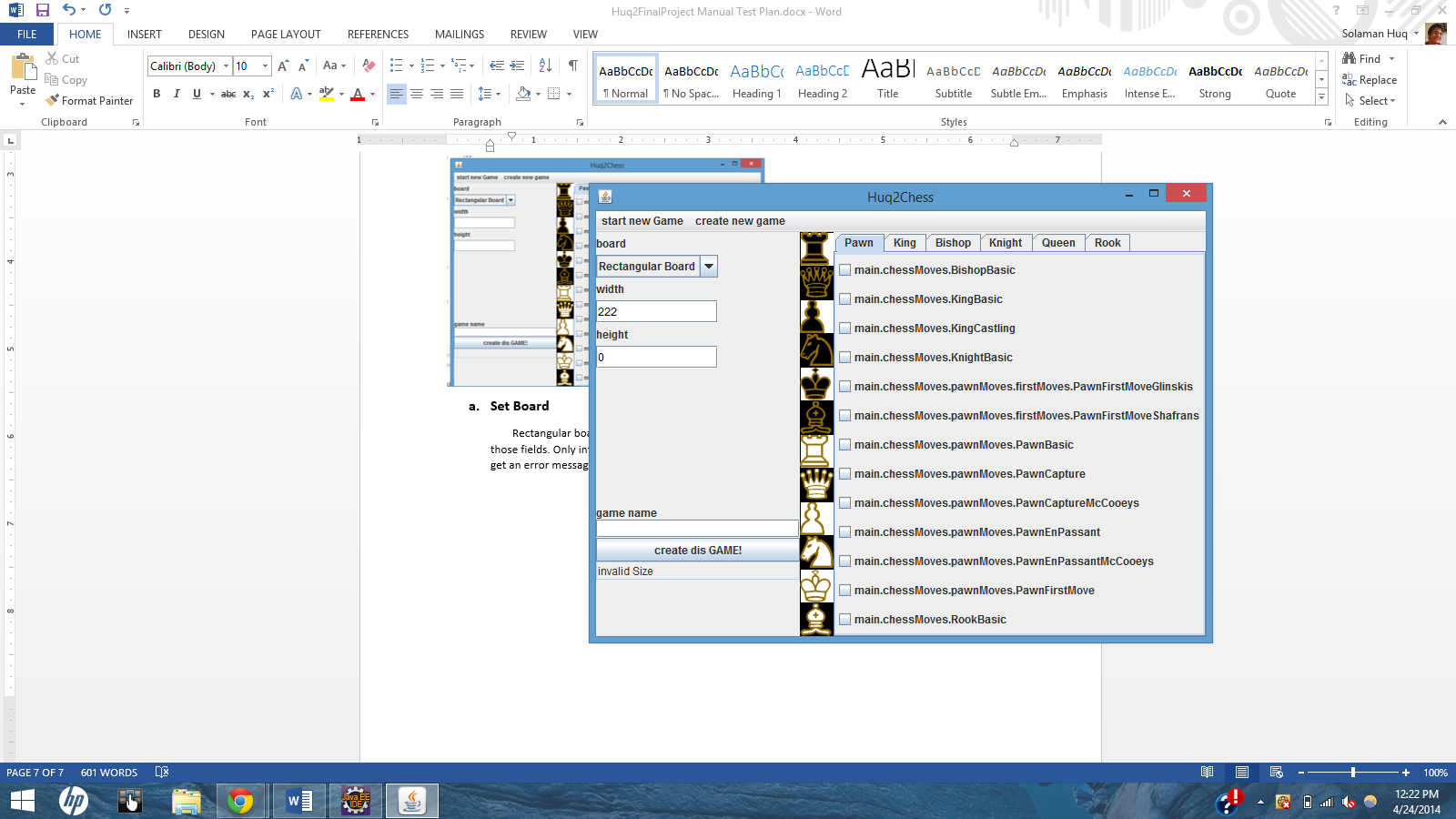
**Custom Games**

Open the Custom game menu, this should be your display

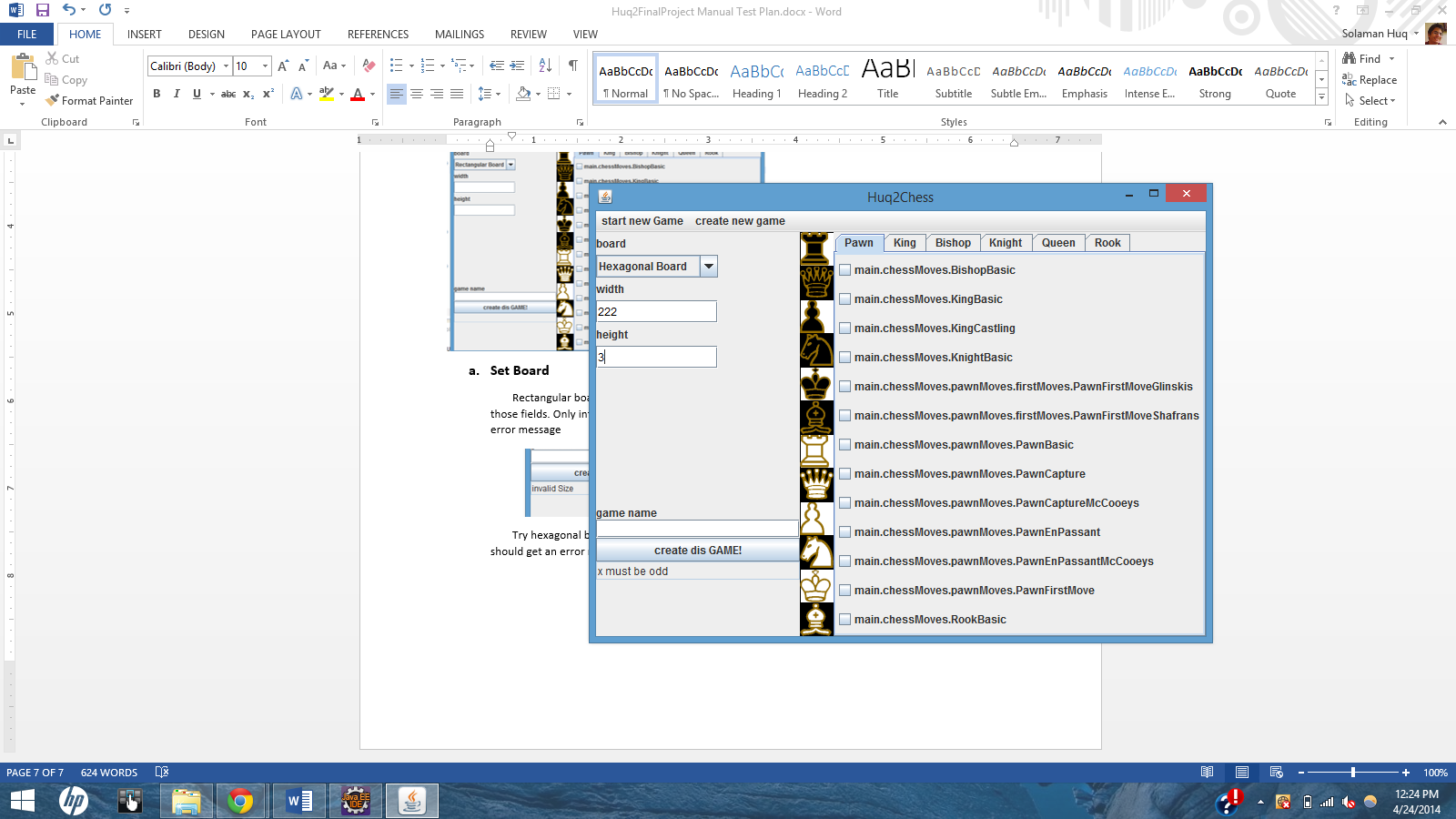


1. **Set Board**

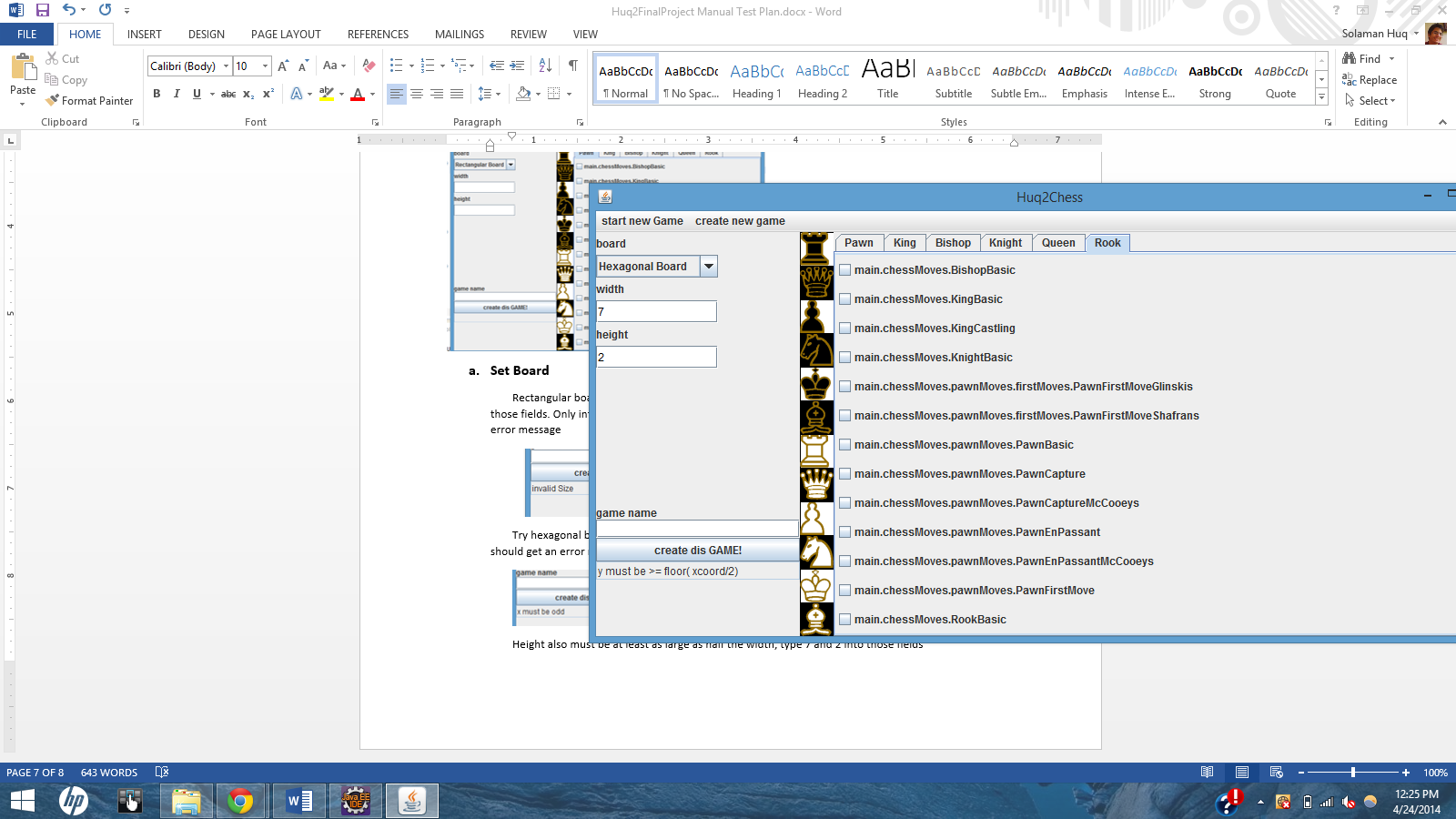
Rectangular boards can have width and height values between 1 and 999. Try typing anything into those fields. Only integers should be displayed. Also, type in 0 and 2 for those fields, you should get an error message



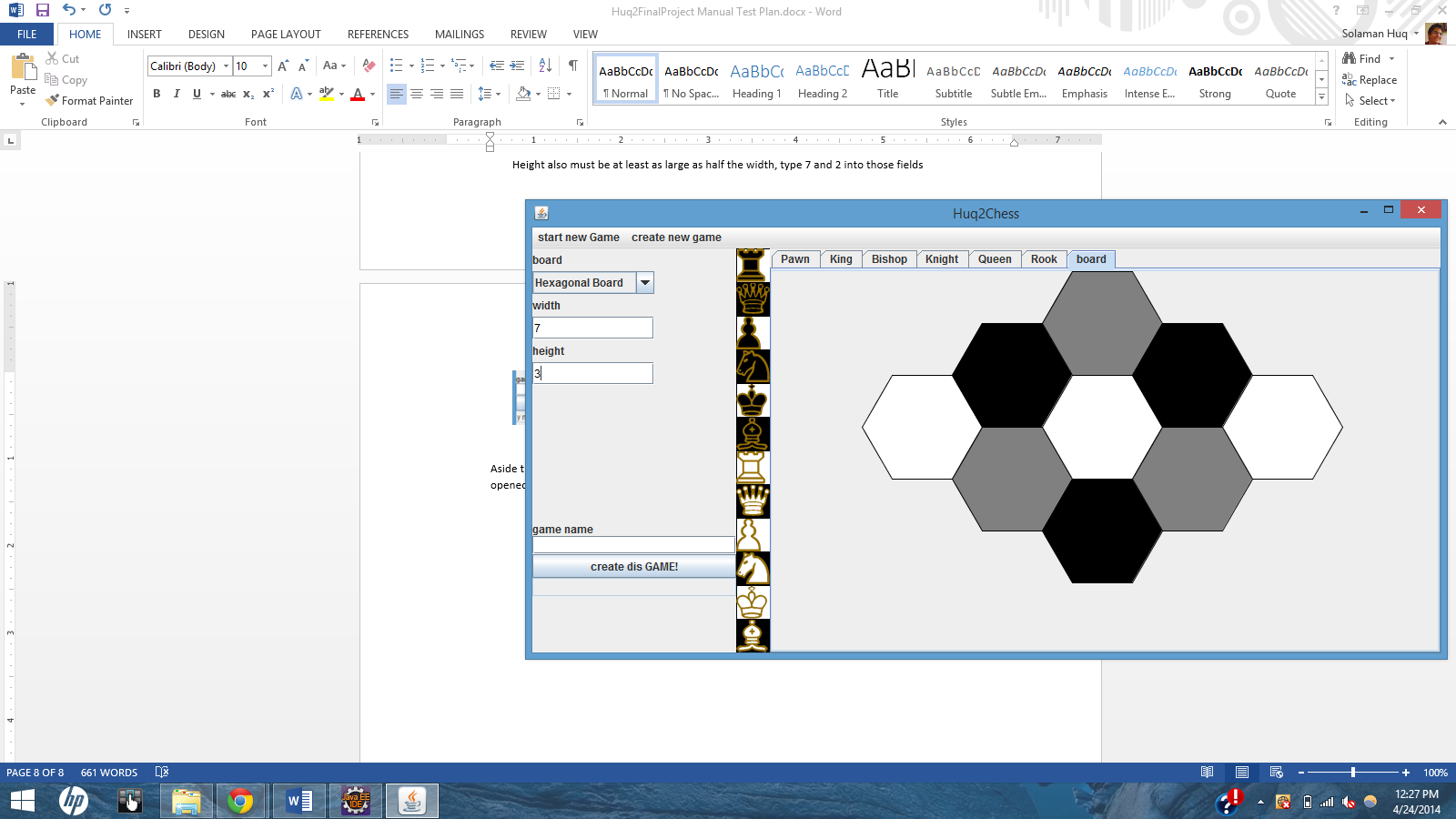
Try hexagonal boards. Width must be odd, try putting in an even value and 7 in the height field. You should get an error message



Height also must be at least as large as half the width, type 7 and 2 into those fields



Aside these conditions, any valid input should automatically generate a board tab that is automatically opened, like so



1. **Laying Pieces**

Try placing pieces from the panel displaying them and then onto the board. Try putting them anywhere else, too. You should get a translucent image of the piece that you can drag. And it will be “set” on the board if you put it on a space.

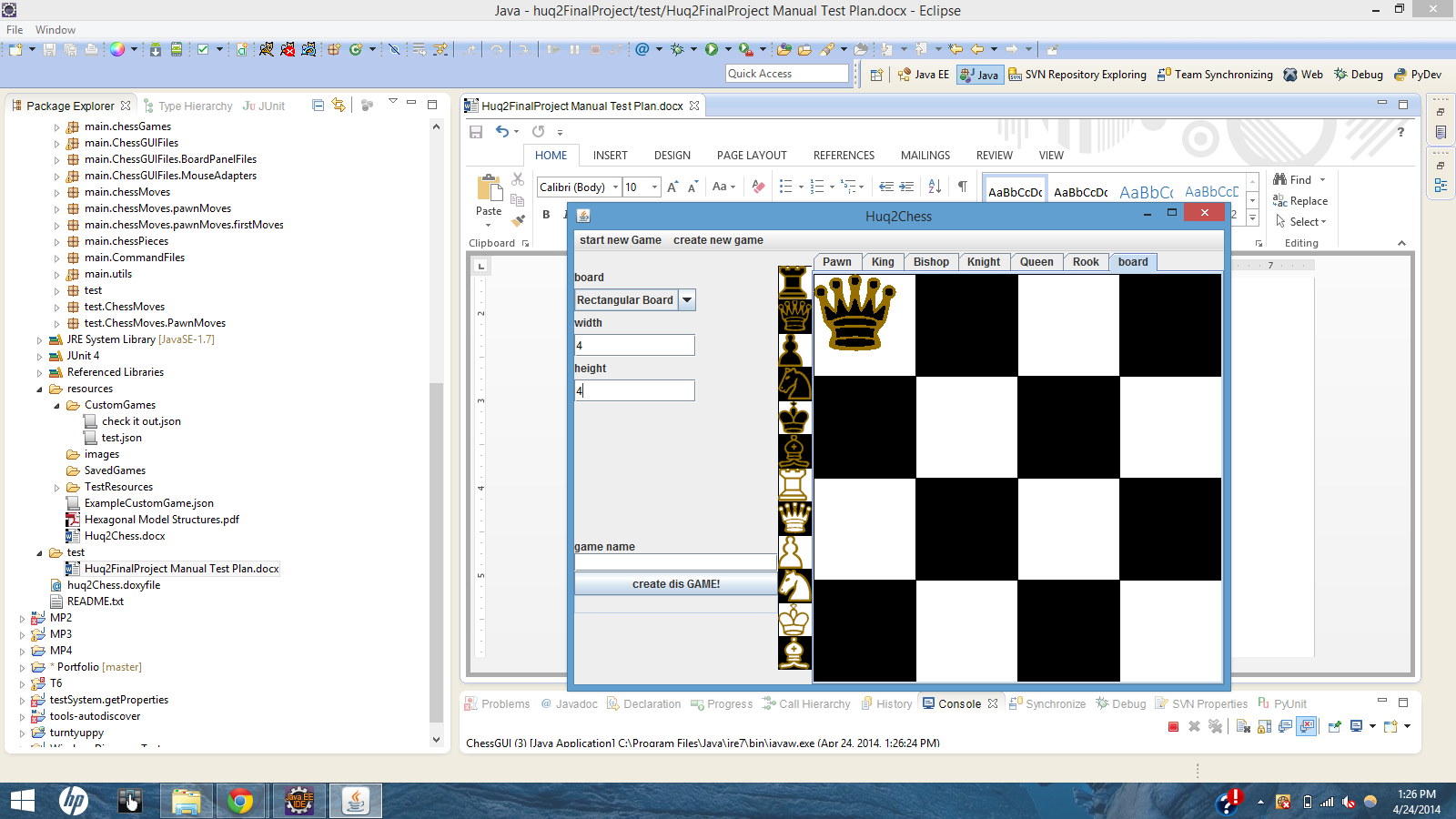
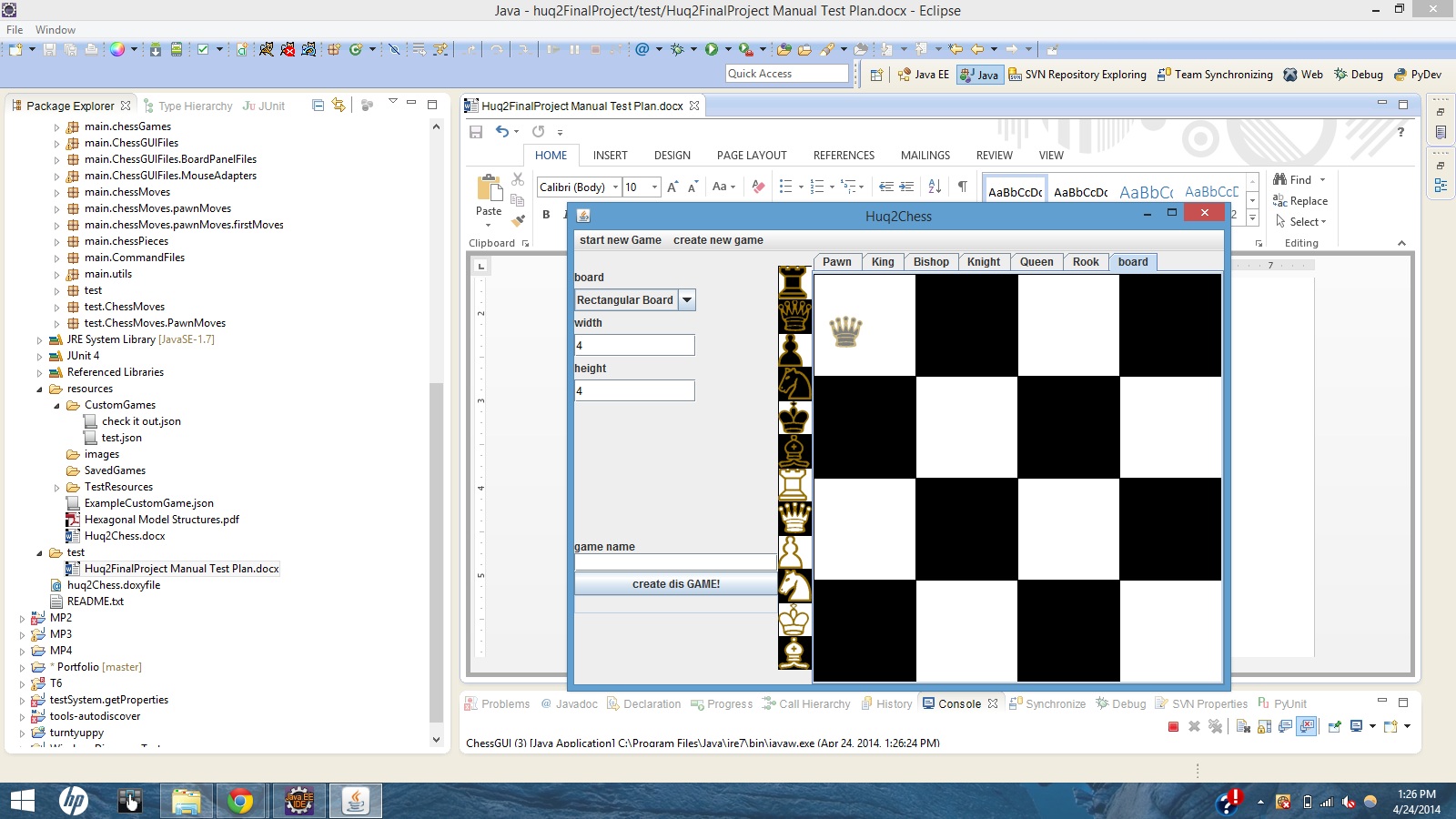
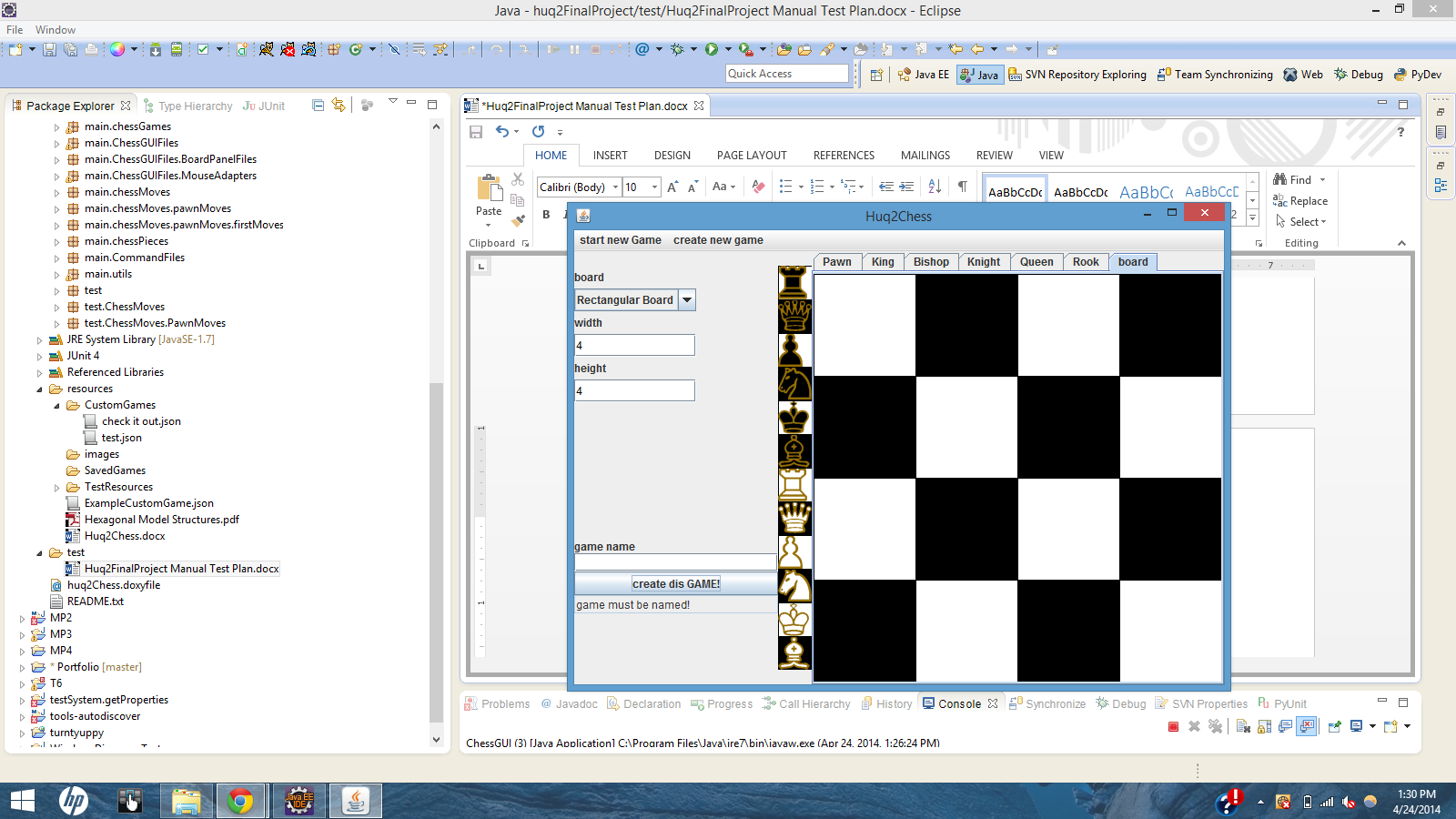


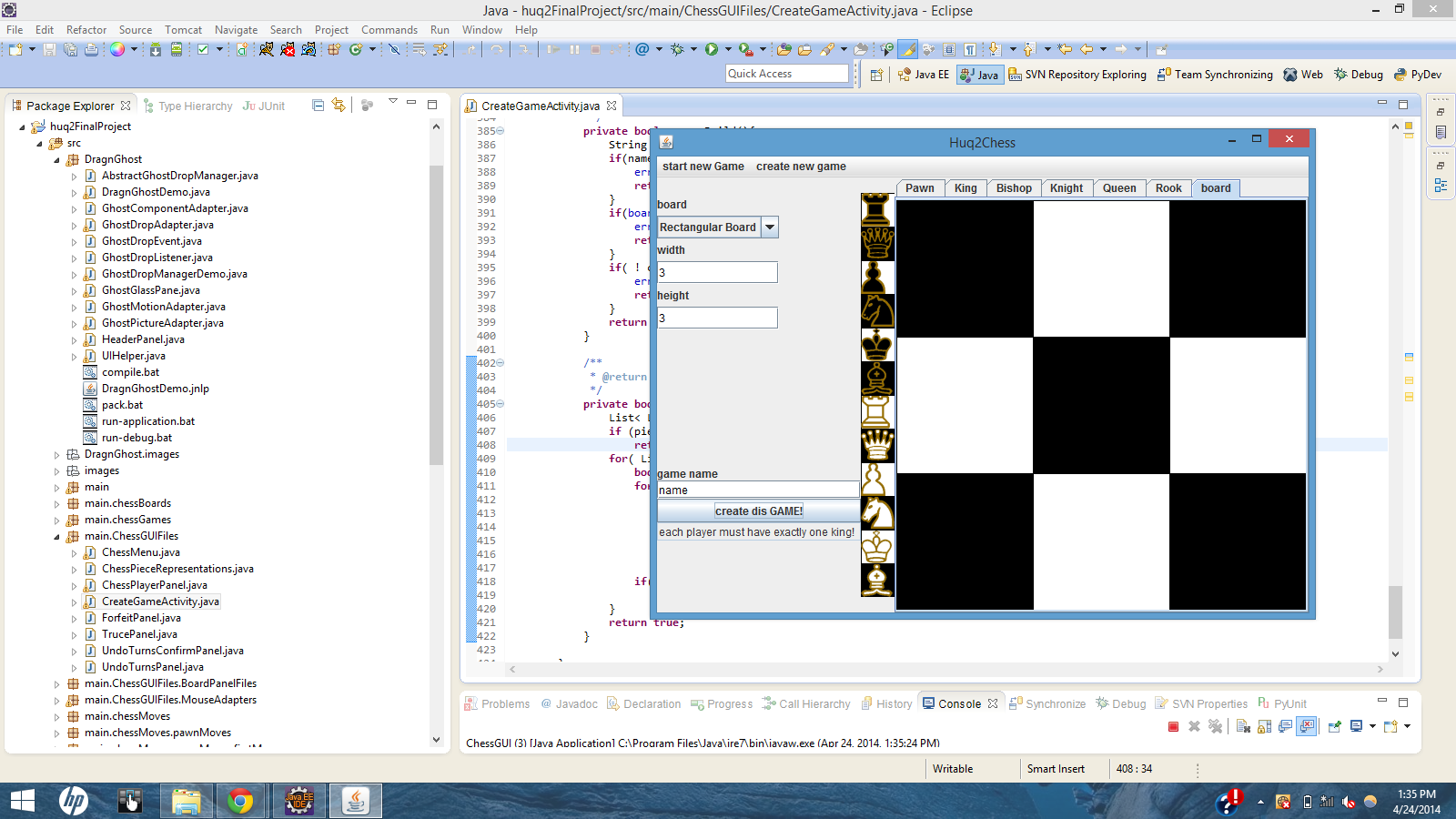
Image of Queen being dragged image of Queen after being dropped

You should also be able to remove a piece by clicking on it in the board and moving the icon away from the board. You should also be able to place the piece back on the board.

1. **Creating the Game**
   * 1. **Without a Name**- Try creating the game *without a name* by pressing the “create dis GAME!” button. You should see an error message.



* + 1. **Without Kings**- Give the game a name, you should only be allowed to type characters and numbers, try the “create dis GAME!” button again. You should see an error message



* + 1. Add Kings to the board, *do not create the game yet.*

1. **Load Outs and Selecting the Game**

You are allowed to add whatever load-out you like to pieces. For testing purposes, go to the “King” tab and select the “main.chessMoves.RookBasic” option. Now Create the game.

Choose the game you just created from the “Start Game” menu. You should see your game, and the king should have available spaces that would be made by a RookBasic ChessMove, like so

