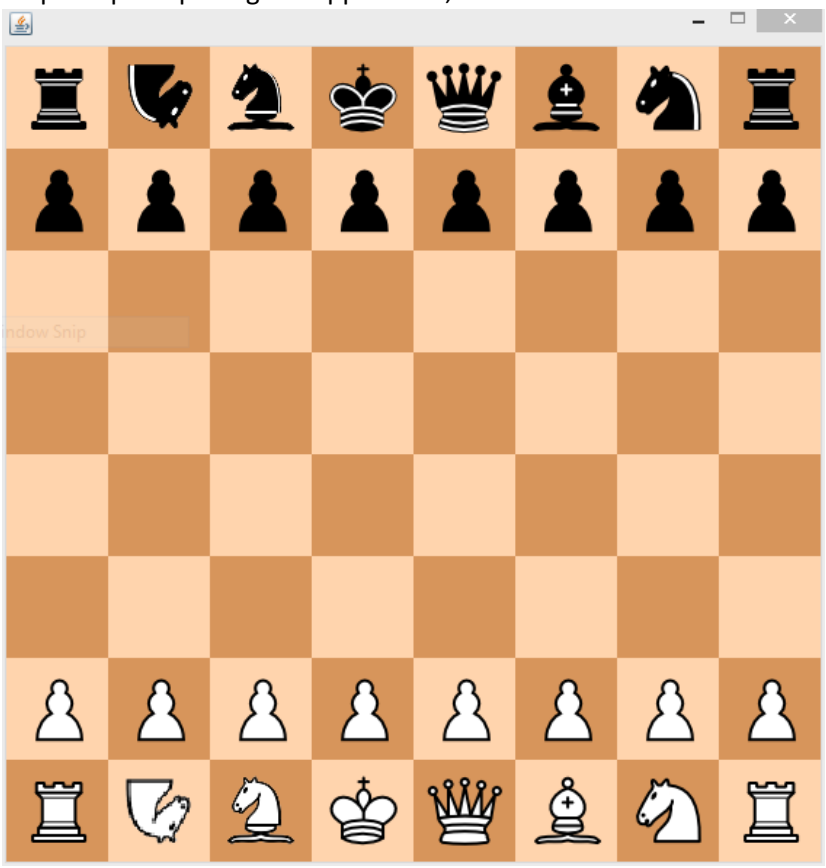
Assignnment1.2: GUI Test Script

Instructions: In order to run the following test cases, compile and run the “Game.java” file as a Java application. In the Eclipse IDE, the Run game button can be pressed, as well as right clicking on “Game.Java” and select “Run as Java Application.”

Test Suite:

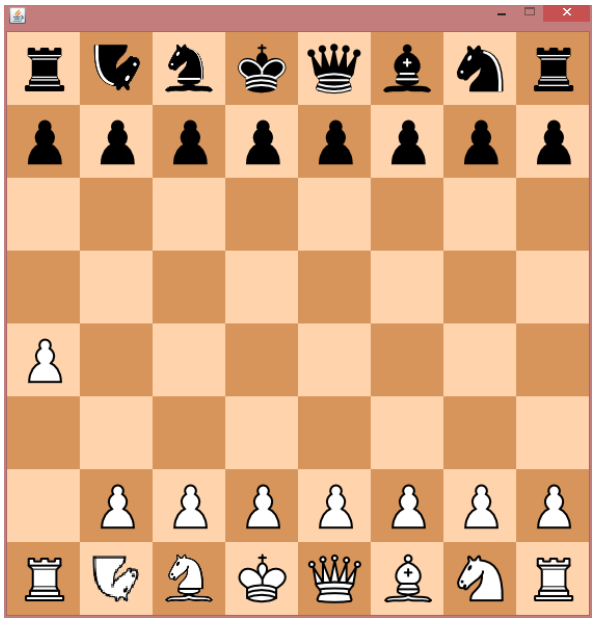
1. Test Simple Functionality
   - ➔ Step 1: Upon opening the application, ensure that the initial screen looks like the following:



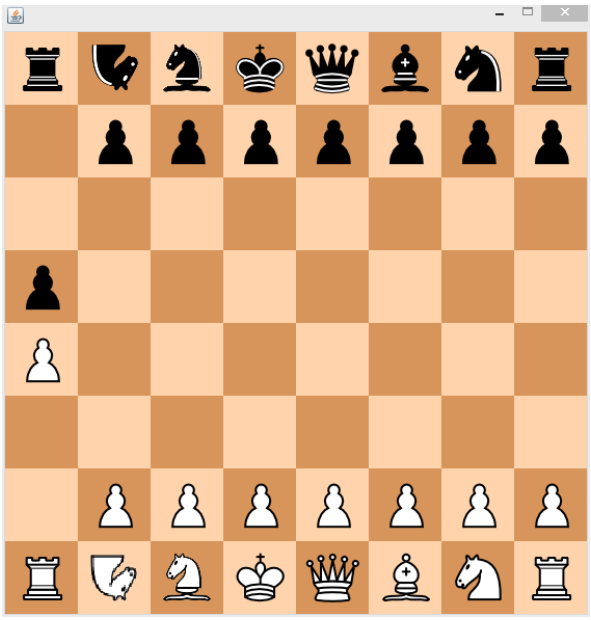
   - ➔ Step 2: Also make sure that an additional window also opens. This window may spawn behind the main chess screen, so you may have to drag the chess screen over.

➔ Step 3: If the two windows matched these pictures, then the application is rendering properly.

2. Test Simple Movement and Movement Restriction
   ➔ Step 1: Rerun the program and move the leftmost white pawn up two spaces by dragging it forward two spaces. The screen should look like this afterwards.



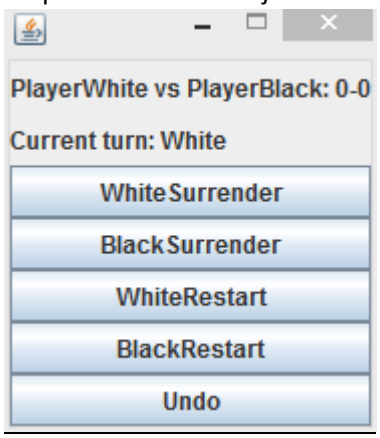
   ➔ Step 2: Move the leftmost black pawn down two spaces using the same dragging motion used to move the white pawn. The screen should look like this afterwards.



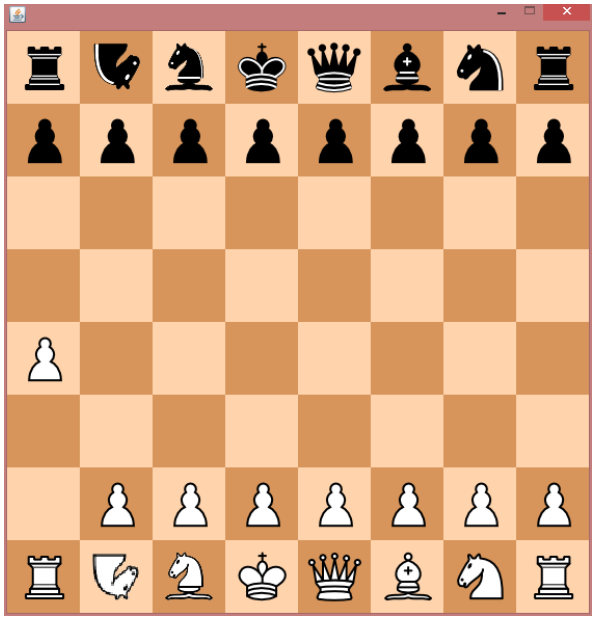
   ➔ Step 3: Make sure that the leftmost white pawn cannot move into the black pawn by dragging the left pawn into the black pawn's position. Nothing should happen. This shows that basic movement and collision detection is working.

3. Test Turn Alternation

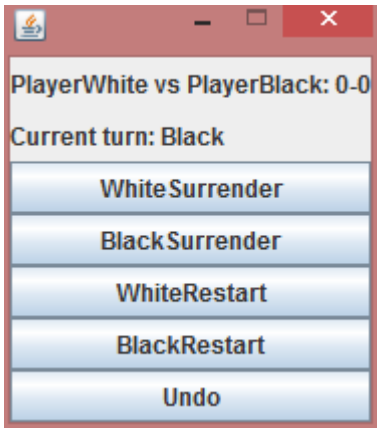
➔ Step 1: Rerun “Game.java” and verify that the periphery screen matches this



➔ Step 2: Move the leftmost white pawn two spaces by dragging it forward two spaces. Again your main screen should look like this.



➔ Step 3: Check that the periphery screen displays “Current Turn: Black”. It should look like this



➔ Step 4: Move the leftmost black pawn down two spaces as in test 2 step 2. Check that the periphery screen shows “Current turn: White”. It should look exactly like how it did in step 1, showing the alternation of turns.

4. Test Surrender

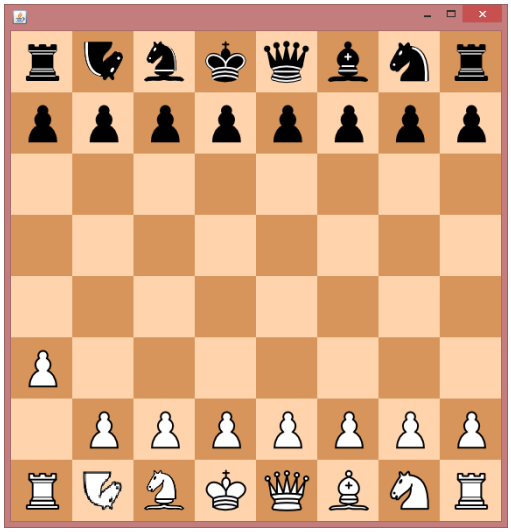
➔ Step 1: Rerun “Game.java”. On the periphery screen, press the “WhiteSurrender” button. A message should pop up saying “Black has Won!” Press Enter.

➔ Step 2: Verify that in the periphery window displays “PlayerWhite vs PlayerBlack: 0-1” at the very top.

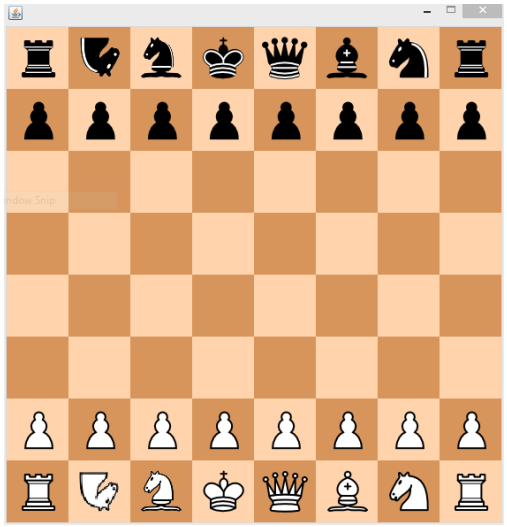
➔ Step 3: Now press the “BlackSurrender” button in the periphery window. A message should pop up saying “Black has Won!” Press Enter and confirm that the periphery window displays “PlayerWhite vs PlayerBlack: 1-1” at the very top.

This confirms the surrendering of games, as well as the incrementing of score between the players.

5. Test Restart.
   - ➔ Step 1: Rerun “Game.java”. Move the leftmost white pawn one space up by dragging it one space forward. The main screen should look like this.



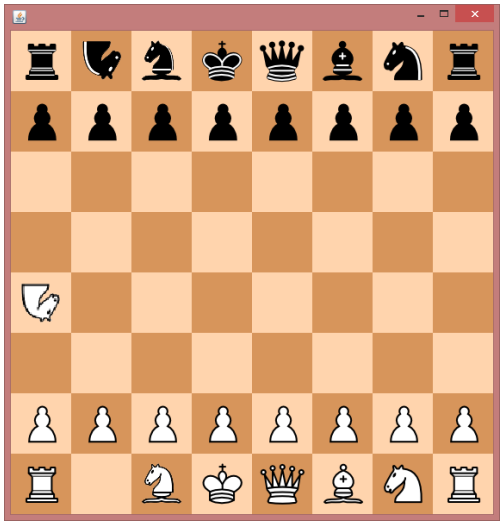
   - ➔ Step 2: Press just the “WhiteRestart” button and confirm that the main screen has not changed.
   - ➔ Step 3: Now press the “BlackRestart” button and confirm that the main screen now looks like a new game.



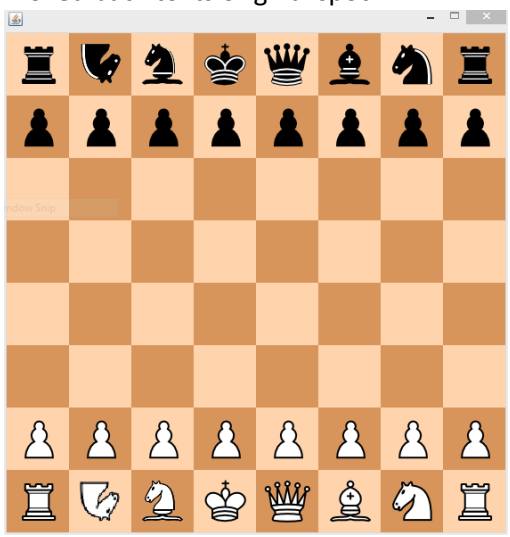
➔ Step 4: Verify that in the periphery window now displays “Current turn: White”, indicating that it is the start of a new game and white starts.

6. Test Undo.

➔ Step 1: Rerun “Game.java”. Move the white camel in the bottom left (signified by an upside-down horse) from its starting position to two spaces ahead of the leftmost pawn.

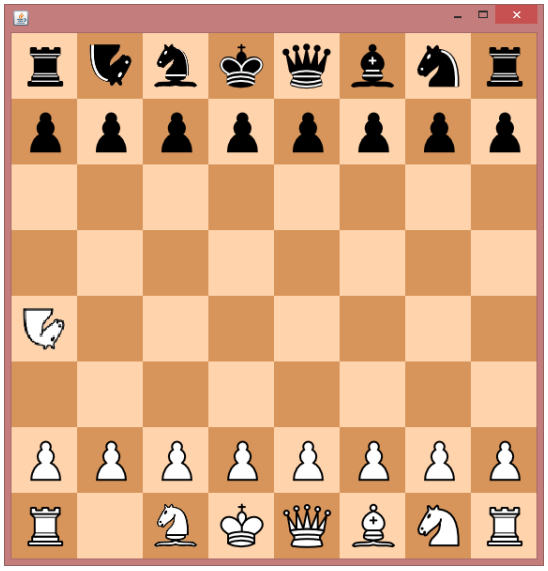


➔ Step 2: Go to the periphery window and press the “Undo” button. Verify that the camel has moved back to its original spot.



➔ Step 3: Now again move the camel back to where it was, two spaces above the white pawn. Now also move the black camel down 3 spaces and 1 to the left, directly in front of the white camel.

➔ Step 4: Now press the “Undo” button and verify that the black camel has moved back to its original spot.



➔ Step 5: Now press the “Undo” button again (or multiple times), and verify that no changes occur, showing that the undo functionality is restricted to the most recent move.