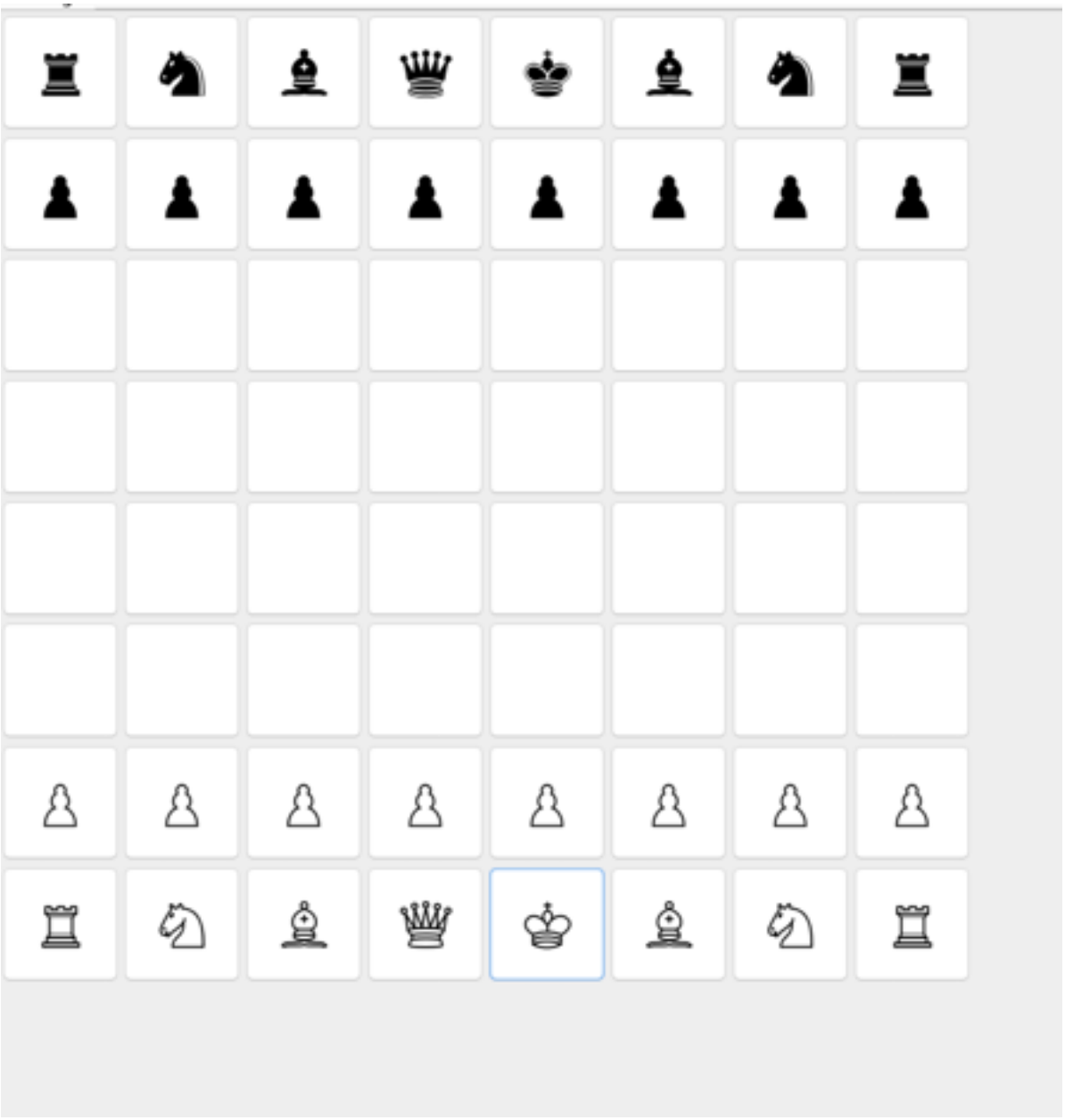
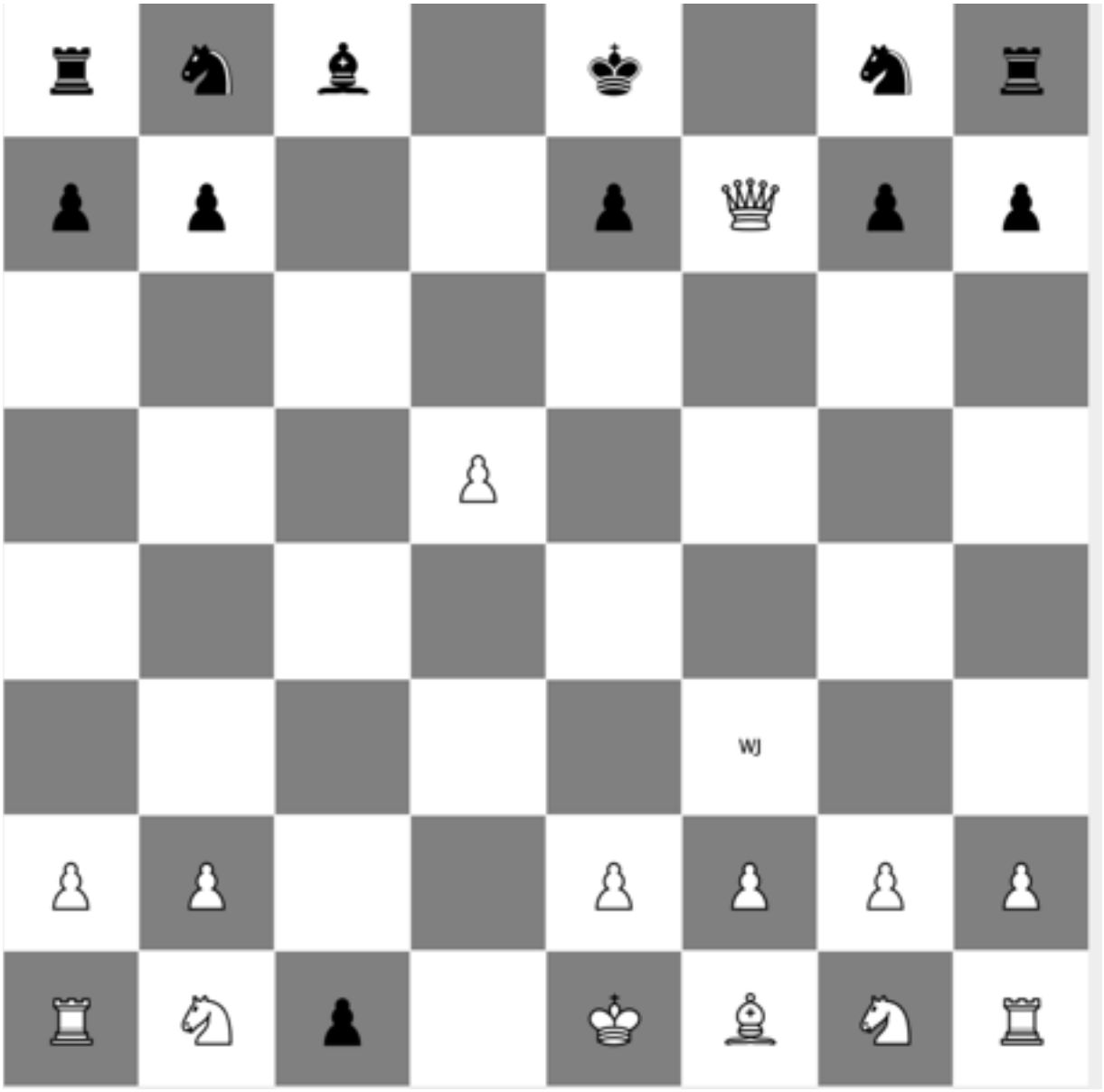
## Manually Test Plan for 1.2:

In the first place, we should test if all the buttons are set correct. The way of doing this is to click on each buttons on the board, and check the information printed out.






And we will see some information like this. By checking this information, we will be sure that the mapping between our Chessboard and our board is correct, which fulfills the MVC models.

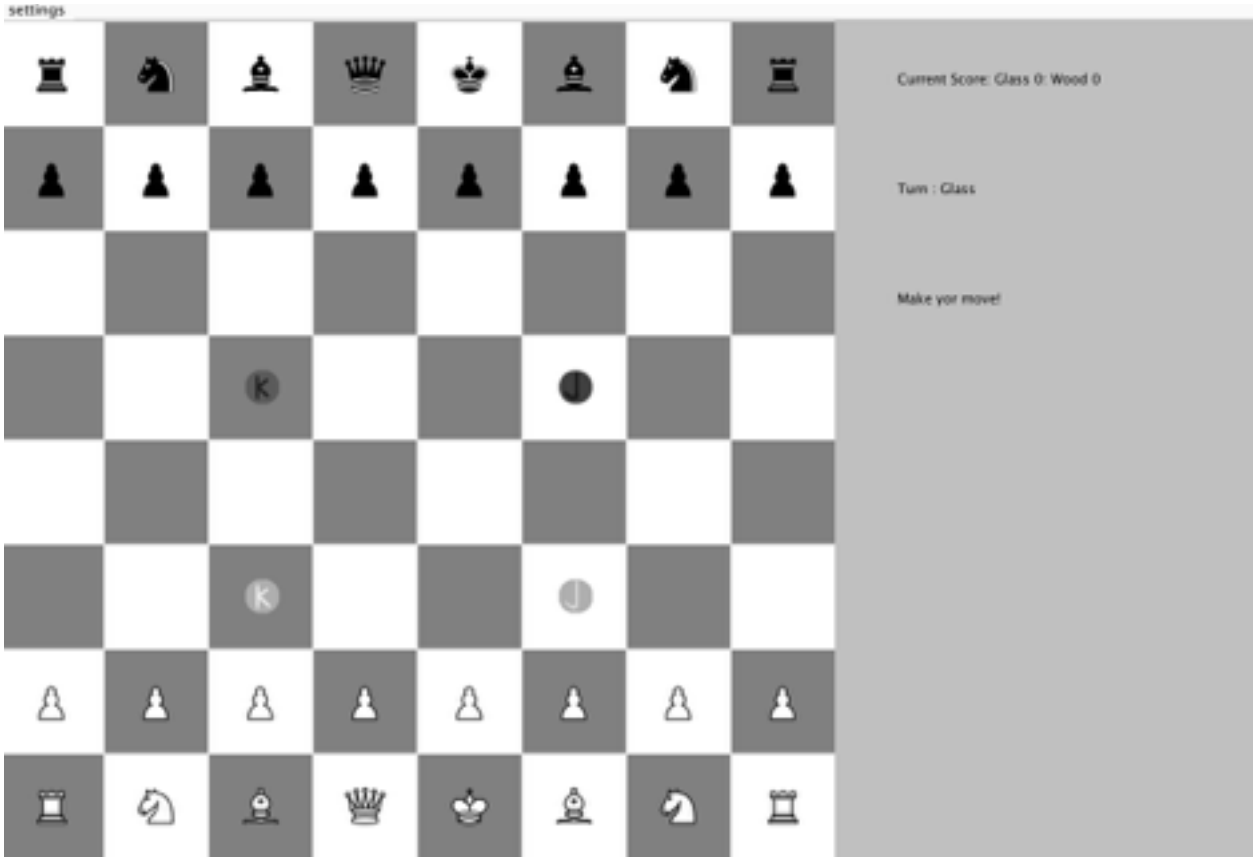
```
}

@Override
public void actionPerformed(ActionEvent e) {
    System.out.println("you have pressed " + x + " " + y);
    if (board.getSquare(new Location(x, y)).hasPiece())
        System.out.println("It is a " + board.getChessPiece(new Location(x, y)).getClass().getSimpleName());
}
```

```
you have pressed 1 1
you have pressed 2 1
you have pressed 3 1
you have pressed 4 1
you have pressed 5 1
```

```
you have pressed 4 2
you have pressed 1 4
you have pressed 2 7
It is a Pawn
you have pressed 2 8
It is a Knight
```

Secondly, we should check if the information given from the game is correct.

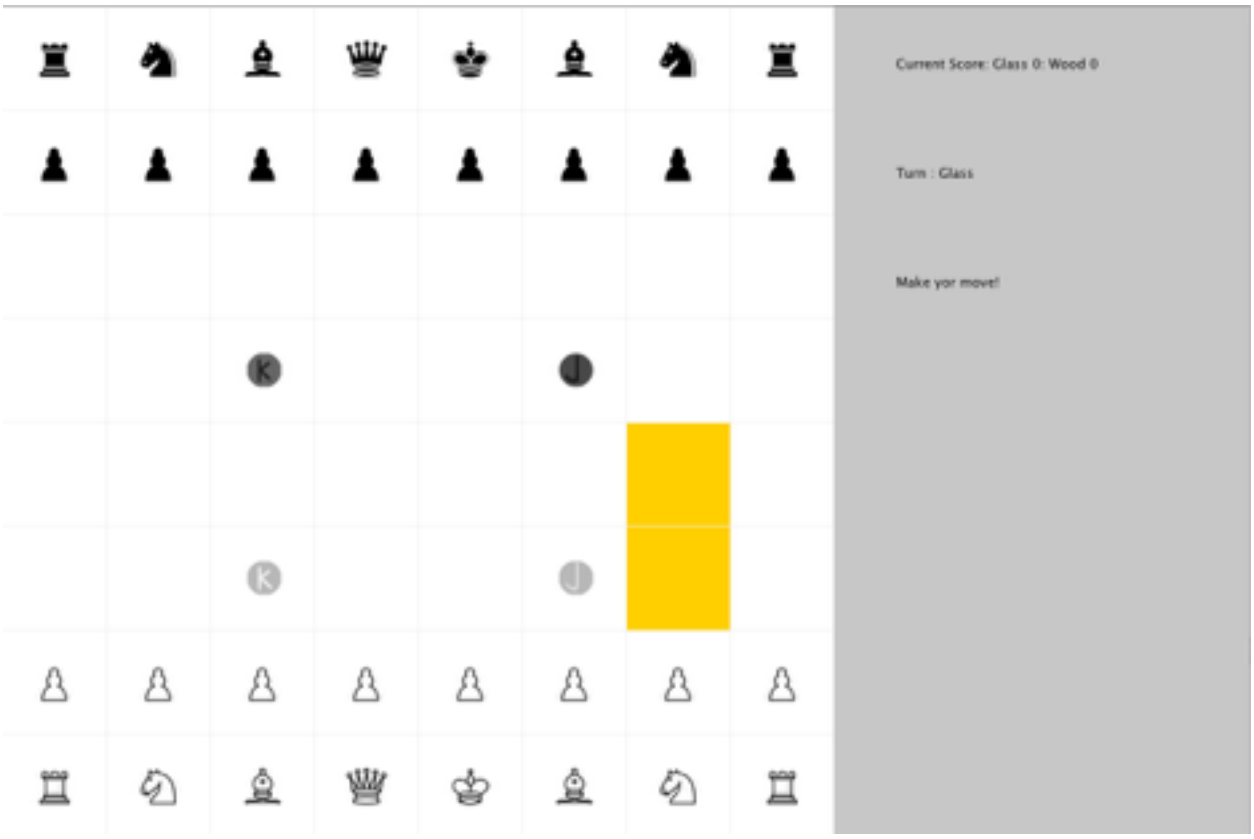
In the right panel we should find the information panel as below.

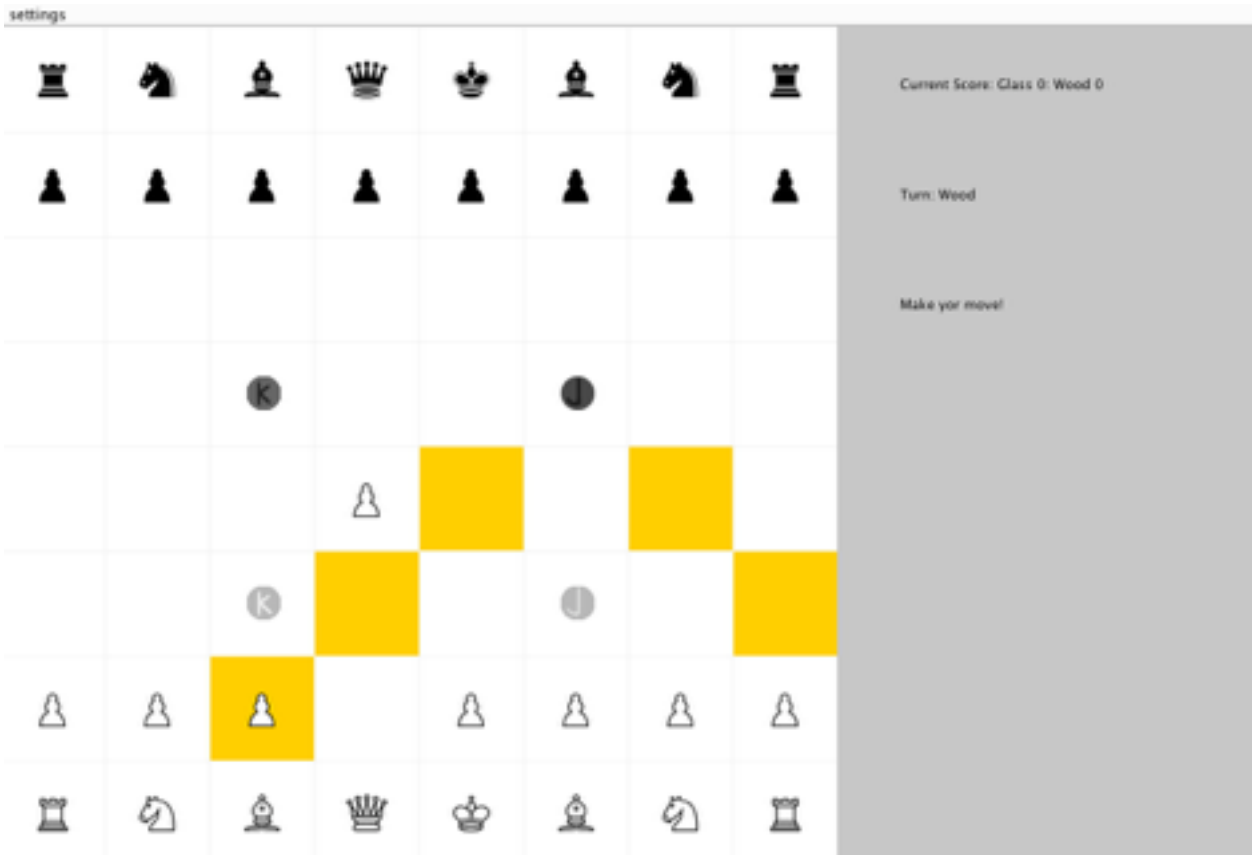


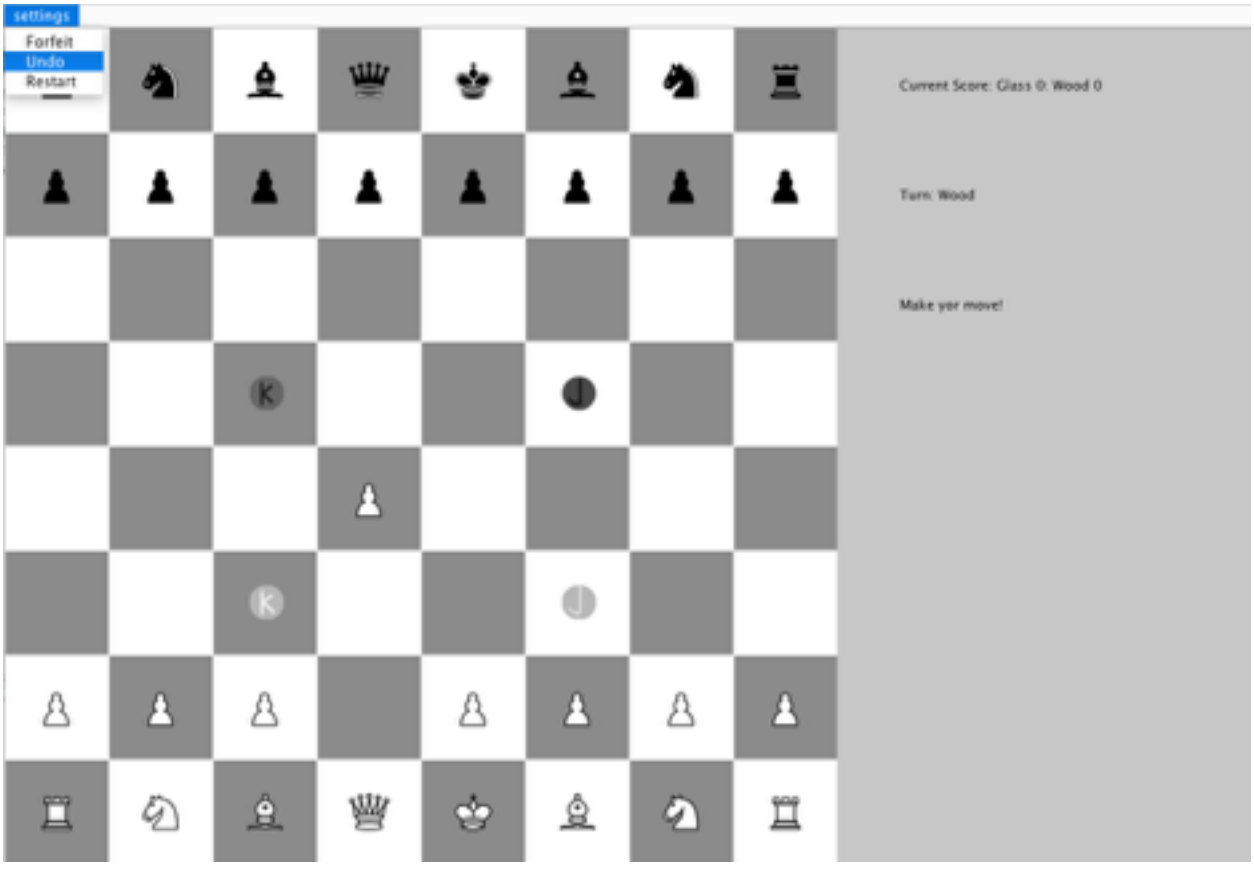
The first line of the information is the current score, the second line is the current turn, and the third line is the information from the system. We need to make sure these informations are correct.

We should check that whenever we have made a move, the turn will change, and when we undo a step, the turn will change back. After several turn and have tested after pressing forfeit and restart, one can be sure that the turn is always correct.

Thirdly, I have implemented the function that when you hover over a button, all its possible moveTO positions will be highlighted with orange color, so that one can be sure about the steps they are going to make. Therefore, we should also check its functionality. We need to be sure that all possible steps are correct: that is, we need to hover over all possible cheeses and check if all of them fit the expected positions. I have tested it in JUnit, but another person should also check if they match the pattern.

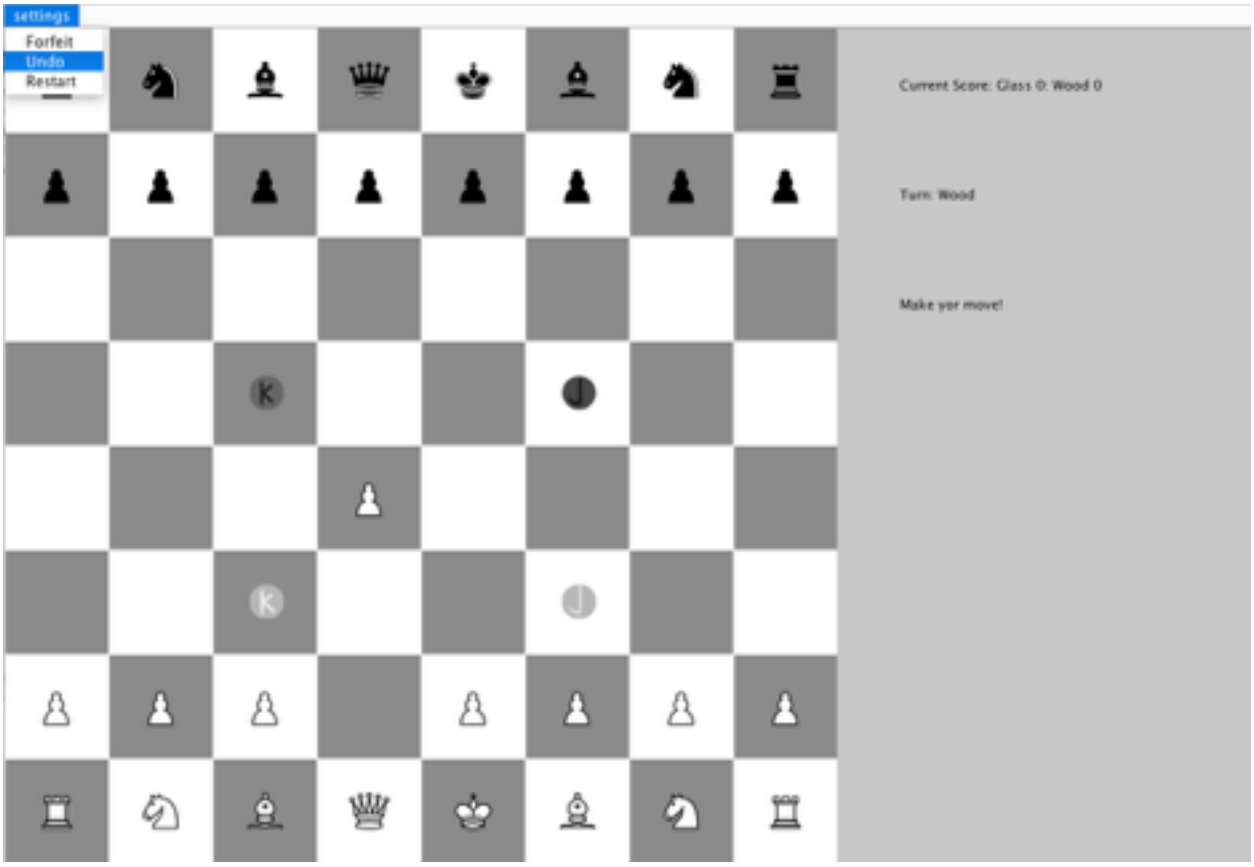



settings
Current Score: Glass 0: Wood 0
Turn: Wood
Make yor move!

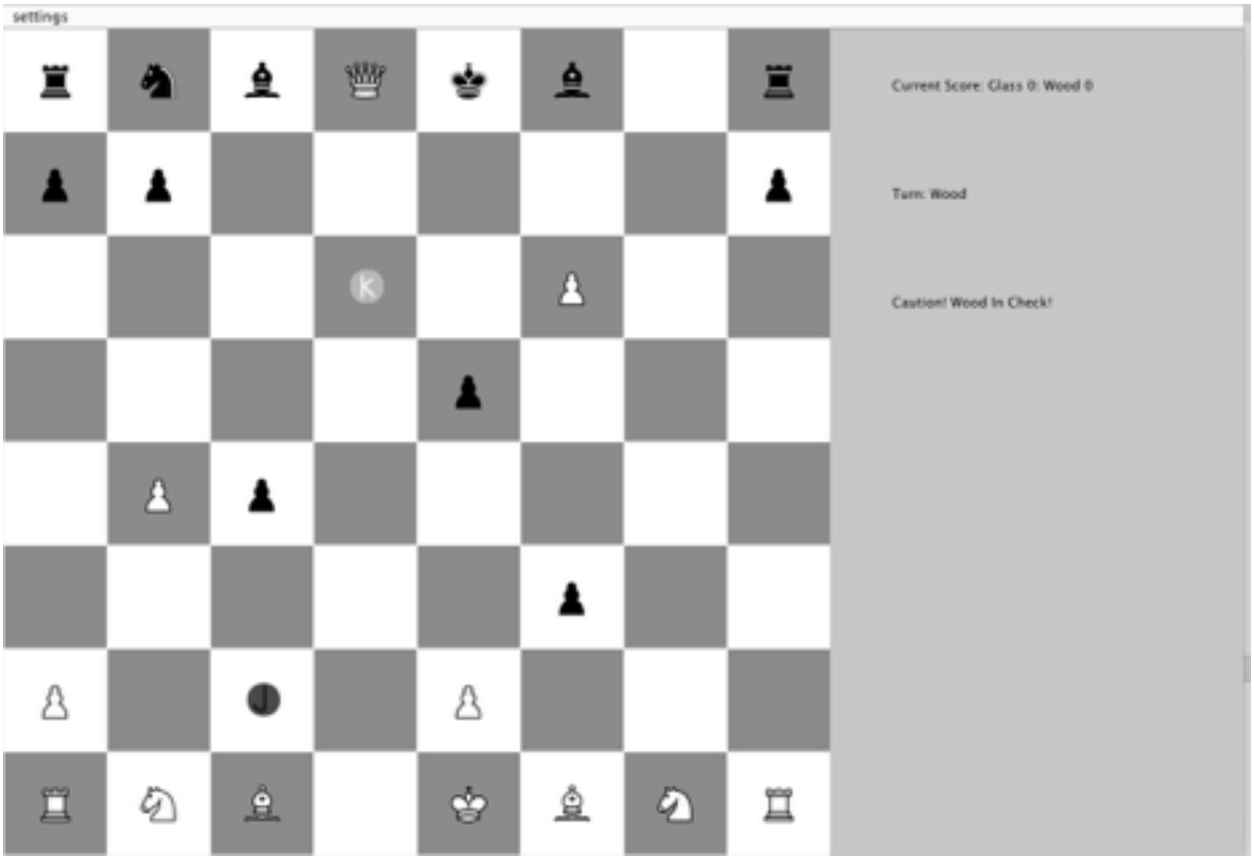

settings
Forfeit
Undo
Restart
Current Score: Glass 0: Wood 0
Turn: Wood
Make yor move!

After checking the routes and turn, we should then check whether Undo has the same visual effect as we have expected.

We should verify the undo function in two situation, where we have captured some chess and we have simply moved to another position.  We need to test them separately.



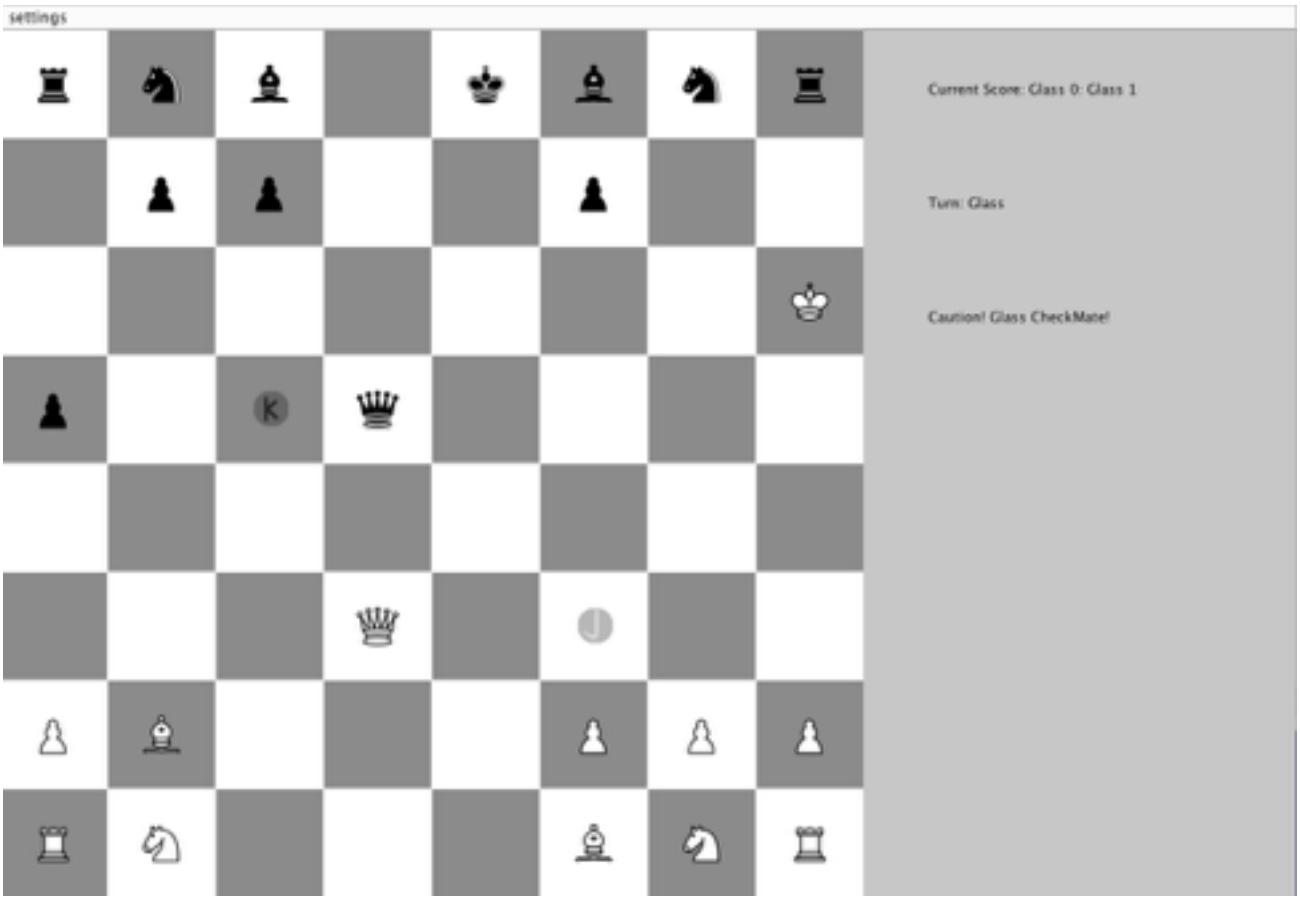

Finally, we should test the InCheck information. When a King is In Check, there will be information shown on the panel, for instance, the following situation is a In Check.




The picture shown above is a case where the Wood(Black) king is in check. We need to check other situations.

We should also check the CheckMate situation, where it is impossible for the player to escape, so we should test the following situation to see it is really a checkMate.

After this, the manual test can be finished.


settings
Current Score: Glass 0: Glass 1
Turn: Glass
Caution! Glass CheckMate!