

```
1 package tests;
2
3 import static org.junit.Assert.*;
4
5 import java.util.Map;
6
7 import org.junit.Assert;
8 import org.junit.BeforeClass;
9 import org.junit.Test;
10
11 import clueGame.Board;
12 import clueGame.DoorDirection;
13 import experiment.BoardCell;
14
15 public class configTest
16 {
17     public static final int LEGEND_SIZE = 11;
18     public static final int NUM_ROWS = 21;
19     public static final int NUM_COLUMNS = 23;
20
21     private static Board board;
22
23     @BeforeClass
24     public static void setUp() {
25         // Board is singleton, get the only instance
26         board = Board.getInstance();
27         // set the file names to use my config files
28         board.setConfigFiles("Clue_Board_Layout_Langfield_Prather.csv",
"LegendForClueLayout.txt");
29         // Initialize will load BOTH config files
30         board.initialize();
31     }
32 }
33
34 @Test
35 public void testRooms()
36 {
37
38     Map<Character, String> legend = board.getLegend();
39     assertEquals(LEGEND_SIZE, legend.size());
40
41     assertEquals("Kindnessroom", legend.get('K'));
42     assertEquals("RepositoryRoom", legend.get('R'));
43     assertEquals("Library", legend.get('L'));
44     assertEquals("Observatory", legend.get('O'));
45     assertEquals("Jumping Jack Room", legend.get('J'));
46     assertEquals("Punching Room", legend.get('P'));
47     assertEquals("Gaming Room", legend.get('G'));
48     assertEquals("Xylophone Room", legend.get('X'));
49     assertEquals("Sexy Room", legend.get('S'));
50
51     assertEquals("Walkway", legend.get('Y'));
52     assertEquals("Closet", legend.get('Q'));
53 }
54
55 @Test
56 public void testBoardDimensions()
57 {
58     // We may have an index error off by one
59     clueGame.BoardCell room = board.getCellAt(8,3);
```

```
60     assertTrue(room.isDoorway());
61     assertEquals(DoorDirection.DOWN, room.getDoorDirection());
62
63
64
65
66     room = board.getCellAt(12,18);
67
68     assertTrue(room.isDoorway());
69     assertEquals(DoorDirection.UP, room.getDoorDirection());
70
71     room = board.getCellAt(20,7);
72     assertTrue(room.isDoorway());
73     assertEquals(DoorDirection.LEFT, room.getDoorDirection());
74
75     room = board.getCellAt(18,3);
76     assertTrue(room.isDoorway());
77     assertEquals(DoorDirection.RIGHT, room.getDoorDirection());
78
79     room = board.getCellAt(1,0);
80     assertTrue(room.isWalkway());
81
82 }
83
84 @Test
85 public void testNumberOfDoors()
86 {
87     int numDoors = 0;
88
89     for (int row=0; row<board.getNumRows(); row++)
90         for (int col=0; col<board.getNumColumns(); col++) {
91             clueGame.BoardCell cell = board.getCellAt(row, col);
92             if (cell.isDoorway())
93                 numDoors++;
94         }
95     Assert.assertEquals(10, numDoors);
96 }
97
98 @Test
99 public void testRoomInitials()
100 {
101     assertEquals((Character)'Y', board.getCellAt(14, 0).getInitial());
102     assertEquals((Character)'Y', board.getCellAt(14, 6).getInitial());
103
104     assertEquals((Character)'Y', board.getCellAt(9, 20).getInitial());
105     assertEquals((Character)'X', board.getCellAt(17, 20).getInitial());
106
107
108 }
109
110
111
112 private Object character(Character initial) {
113     // TODO Auto-generated method stub
114     return null;
115 }
116
117 }
118
```