

D'BUGGERS



BATTLESHIP

THE BATTLE FOR EARTH BEGINS

WWW.DESIGNPANOPLY.COM

D'Buggers

Battleship Game

JAVA BATTLESHIP - ** Yuval Ma

Player SETUP:

	1	2	3	4	5	6	7	8	9	10
A	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

Ship #1: Length-2
Type in row (A-J):

Ship #1: Length-2
Type in row (A-J): a
Type in column (1-10): 1
Type in direction (0-H, 1-V): 0

	1	2	3	4	5	6	7	8	9	10
A	D	D	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

You have 4 remaining ships to place.

Ship #2: Length-3
Type in row (A-J):

** USER MISS AT A1 **

USER MAKE GUESS:
Viewing My Guesses:

	1	2	3	4	5	6	7	8	9	10
A	0	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

Type in row (A-J):

YOUR BOARD...PRESS ENTER TO CONTINUE...

	1	2	3	4	5	6	7	8	9	10
A	D	X	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	B
D	-	-	-	-	-	-	-	-	-	B
E	-	-	-	C	C	C	-	-	-	B
F <td>C</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>B</td>	C	-	-	-	-	-	-	-	-	B
G <td>C</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td>	C	-	-	-	-	-	-	-	-	-
H <td>C</td> <td>-</td> <td>-</td> <td>X</td> <td>X</td> <td>A</td> <td>A</td> <td>A</td> <td>-</td> <td>-</td>	C	-	-	X	X	A	A	A	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

PRESS ENTER TO CONTINUE...

** USER MISS AT I1 **

USER MAKE GUESS:
Viewing My Guesses:

	1	2	3	4	5	6	7	8	9	10
A	0	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	0	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	0	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	0	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	0	-	-	-	-	-	-	-	0	-
J	-	-	-	-	-	-	-	-	-	-

Type in row (A-J): |

Type in column (1-10): 1

COMPUTER IS MAKING GUESS...
COMP HIT AT A2

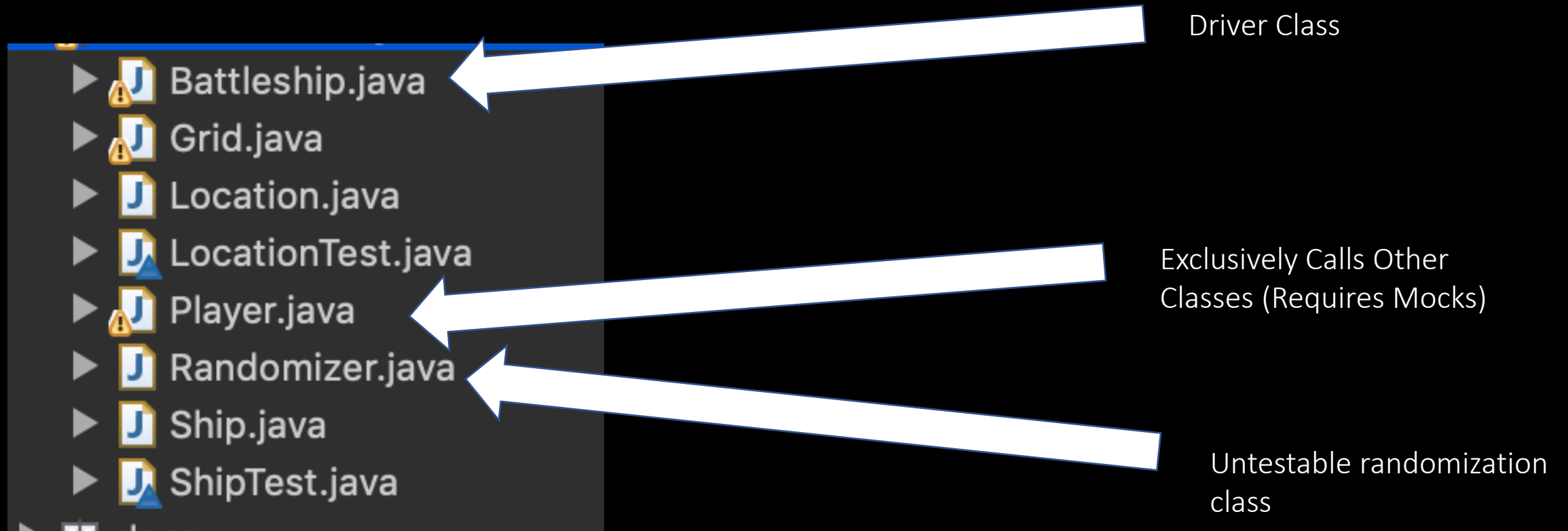
YOUR BOARD...PRESS ENTER TO CONTINUE...

	1	2	3	4	5	6	7	8	9	10
A	D	X	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	B
D	-	-	-	-	-	-	-	-	-	B
E	-	-	-	C	C	C	-	-	-	B
F <td>C</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>B</td>	C	-	-	-	-	-	-	-	-	B
G <td>C</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td>	C	-	-	-	-	-	-	-	-	-
H <td>C</td> <td>-</td> <td>-</td> <td>X</td> <td>X</td> <td>A</td> <td>A</td> <td>A</td> <td>-</td> <td>-</td>	C	-	-	X	X	A	A	A	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

PRESS ENTER TO CONTINUE...

** USER MISS AT I1 **

Class Structure & Excluded Classes From Test



Unit Tests On Location Class

```
20 // File: LocationTest.java
21 import static org.junit.jupiter.api.Assertions.*;
22
23 //=====
24 // Class declarations
25 //=====
26
27 class LocationTest
28 {
29     // Local constants for this test class
30     static final int UNSET = 0;
31
32     @Test
33     void testLocation()
34     {
35         Location testLoc = new Location();
36         assertEquals(testLoc.getStatus(), 0);
37         assertEquals(testLoc.hasShip(), false);
38         assertEquals(testLoc.getLengthOfShip(), -1);
39         assertEquals(testLoc.getDirectionOfShip(), -1);
40     }
41
42     @Test
43     void testCheckHit()
44     {
45         Location testLoc = new Location();
46         assertEquals(testLoc.checkHit(), false);
47         testLoc.markHit();
48         assertEquals(testLoc.checkHit(), true);
49     }
50
51     @Test
52     void testCheckMiss()
53     {
54         Location testLoc = new Location();
55         assertEquals(testLoc.checkMiss(), false);
56         testLoc.markMiss();
57         assertEquals(testLoc.checkMiss(), true);
58     }
59 }
60
```

Finished after 0.134 seconds

Runs: 10/10 Errors: 0 Failures: 0

Failure Trace

Problems Javadoc Declaration Console Coverage

<terminated> LocationTest [JUnit] /Library/Java/JavaVirtualMachines/jdk-12.0.1.jdk/Contents/Home/bin/java (May 2, 2019, 11:3

```
1 // Location constructor.
2 public Location()
3 {
4     // Set initial values
5     status = 0;
6     hasShip = false;
7     lengthOfShip = -1;
8     directionOfShip = -1;
9 }
10
11 // Was this Location a hit?
12 public boolean checkHit()
13 {
14     if (status == HIT)
15         return true;
16     else
17         return false;
18 }
19
20 // Was this location a miss?
21 public boolean checkMiss()
22 {
23     if (status == MISSED)
24         return true;
25     else
26         return false;
27 }
28
29 // Was this location unguessed?
30 public boolean isUnguessed()
31 {
32     if (status == UNGUESSED)
33         return true;
34     else
35         return false;
36 }
37
38 // Mark this location a hit.
```

Code Coverage Analysis

(default package) (Jun 5, 2019 11:08:44 PM)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
▼ Java-Battleship		1,482	3,099	4,581
▼ Java-Battleship		1,482	3,099	4,581
> (default package)			2,783	2,783
▼ testedFiles	82.4 %	1,482	316	1,798
> Grid.java	49.7 %	304	308	612
> ShipTest.java	96.2 %	203	8	211
> GridTest.java	100.0 %	460	0	460
> Location.java	100.0 %	74	0	74
> LocationTest.java	100.0 %	333	0	333
> Ship.java	100.0 %	108	0	108

Code Coverage on Grid

```
// Return the number of rows in the Grid
public int numRows()
{
    return NUM_ROWS;
}

// Return the number of columns in the grid
public int numCols()
{
    return NUM_COLS;
}

public void printStatus()
{
    generalPrintMethod(0);
}

public void printShips()
{
    generalPrintMethod(1);
}

public void printCombined()
{
    generalPrintMethod(2);
}

public boolean hasLost()
{
    if (points >= 17)
        return true;
    else
        return false;
}
```



```
// Type: 0 for status, 1 for ships, 2 for combined
private void generalPrintMethod(int type)
{
    System.out.println();
    // Print columns (HEADER)
    System.out.print(" ");
    for (int i = 1; i <= NUM_COLS; i++)
    {
        System.out.print(i + " ");
    }
    System.out.println();

    // Print rows
    int endLetterForLoop = (NUM_ROWS - 1) + 65;
    for (int i = 65; i <= endLetterForLoop; i++)
    {
        char theChar = (char) i;
        System.out.print(theChar + " ");

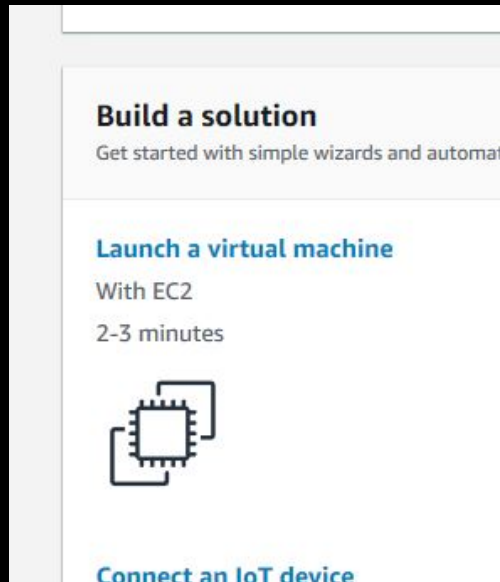
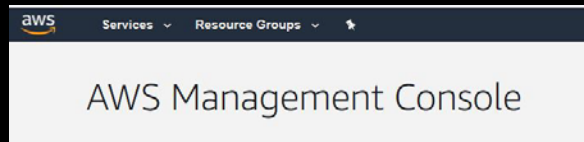
        for (int j = 0; j < NUM_COLS; j++)
        {
            if (type == 0) // type == 0; status
            {
                if (grid[switchCounterToIntegerForArray(i)][j].isUngessed())
                    System.out.print("- ");
                else if (grid[switchCounterToIntegerForArray(i)][j].checkMiss())
                    System.out.print("O ");
                else if (grid[switchCounterToIntegerForArray(i)][j].checkHit())
                    System.out.print("X ");
            }
            else if (type == 1) // type == 1; ships
            {
                if (grid[switchCounterToIntegerForArray(i)][j].hasShip())
                {
                    // System.out.print("X ");
                    if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 2)
                    {
                        System.out.print("D ");
                    }
                    else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 3)
                    {
                        System.out.print("C ");
                    }
                    else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 4)
                    {
                        System.out.print("B ");
                    }
                    else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 5)
                    {
                        System.out.print("A ");
                    }
                }
            }
            else if (!grid[switchCounterToIntegerForArray(i)][j].hasShip())
            {
                System.out.print("- ");
            }
        }
        else // type == 2; combined
        {
            if (grid[switchCounterToIntegerForArray(i)][j].checkHit())
                System.out.print("X ");
            else if (grid[switchCounterToIntegerForArray(i)][j].hasShip())
            {
                // System.out.print("X ");
                if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 2)
                {
                    System.out.print("D ");
                }
                else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 3)
                {
                    System.out.print("C ");
                }
                else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 4)
                {
                    System.out.print("B ");
                }
                else if (grid[switchCounterToIntegerForArray(i)][j].getLengthOfShip() == 5)
                {
                    System.out.print("A ");
                }
            }
            else
            {
                System.out.print("- ");
            }
        }
    }
    System.out.println();
}
```


Stress Testing and Test Suite Execution

```
78
79 echo -e "
80 =====
81 Stress Test Execution
82 =====
83 " | tee -a $LOG_FILE
84
85 numInstances=$1
86 numIter=$2
87
88 # Capture START of test section
89 echo "Start date/time:" $(date) | tee -a $LOG_FILE
90 start=$SECONDS
91
92 # Kick off the background processes
93 for i in `seq 1 $numInstances`; do
94     echo "Launching test suite instance $i of $numInstances..." | tee -a $LOG_FILE
95
96     # This block gets spun off as a non-blocking background thread.
97     # Avoid resource sharing (e.g. writing to same log file) between threads
98     # to avoid performance bottlenecks inside this block.
99     (
100         # Obtain a unique index for this block
101         fileIdx=$i
102
103         # Execute the test suite
104         ./runTestSuite.sh $numIter >> ${LOG_PREFIX}_${fileIdx}.${LOG_EXT}
105     ) &
106
107     # Add block PID to list
108     pidList[$i]=$!
109 done
110
111 # Block until all background tasks are complete - essentially performing a join
112 for i in `seq 1 $numInstances`; do
113     echo "Waiting for instance $i (pid=${pidList[$i]}) to complete..." | tee -a $LOG_FILE
114     wait ${pidList[$i]}
115 done
```

```
117 # Capture END of test section
118 stop=$SECONDS
119 echo "Stop date/time:" $(date) | tee -a $LOG_FILE
120
121 # Combine all output to one file for easier parsing
122 sleep 1 # Allow time for previous file writes to complete
123 for i in `seq 1 $numInstances`; do
124     cat ${LOG_PREFIX}_${i}.${LOG_EXT} >> ${LOG_PREFIX}_combined.${LOG_EXT}
125 done
126
127 # Parse results and report PASS/FAIL
128 expectedNumPass=$numInstances
129 actualNumPass=$(grep PASS -c ${LOG_PREFIX}_combined.${LOG_EXT})
130
131 if [ "$actualNumPass" -eq "$expectedNumPass" ]; then
132     result="PASS"
133     rval=0
134 else
135     result="FAIL"
136     rval=4
137 fi
138
139 # Perform calculations
140 duration=$(( $stop - $start ))
141 passRate=$(bc -l <<< "scale=2; $actualNumPass/$numInstances*100")
142
143 echo "" | tee -a $LOG_FILE
144 echo "===== " | tee -a $LOG_FILE
145 echo "Results & Statistics" | tee -a $LOG_FILE
146 echo "===== " | tee -a $LOG_FILE
147 echo "Overall stress test result:      $result" | tee -a $LOG_FILE
148 echo "Execution time:                  $duration [seconds]" | tee -a $LOG_FILE
149 echo "" | tee -a $LOG_FILE
150 echo "Expected # of passing instances:  $expectedNumPass" | tee -a $LOG_FILE
151 echo "Actual # of passing instances:    $actualNumPass" | tee -a $LOG_FILE
152 echo "Stress test passing rate:         $passRate%" | tee -a $LOG_FILE
153 echo "===== " | tee -a $LOG_FILE
154 echo "Please see the log file for the full console output: $LOG_FILE"
155 echo ""
156
157 # Return status code
158 exit $rval
```

Configuring A Server For Automated Testing

A screenshot of the AWS console showing a list of EC2 instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, and Alarms. One instance is listed with ID i-083ce917d29069468, type t2.micro, in us-east-1b, and is in a 'running' state with 2/2 checks passed.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarms
	i-083ce917d29069468	t2.micro	us-east-1b	running	2/2 checks ...	Non

A screenshot of a terminal window on an EC2 instance. The title bar shows 'ec2-user@ip-172-31-42-109:~'. The terminal output shows the login process for 'ec2-user' using a public key, the last login time, and the Amazon Linux logo. It then shows the command 'nvm --version' being executed, returning '0.33.2'.

```
ec2-user@ip-172-31-42-109:~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
Last login: Tue Aug 29 09:12:08 2017 from 183.82.229.14  
  
 _ _ _ _ _  
 _ | ( _ | /   Amazon Linux AMI  
 _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/  
[ec2-user@ip-172-31-42-109 ~]$ nvm --version  
0.33.2  
[ec2-user@ip-172-31-42-109 ~]$
```



<http://ec2-54-163-130-1.compute-1.amazonaws.com:8080/>

Username: mckeeem
Password: pickleball_2019