Battleship – Boards Guide

Overview

The project supplies an initial set of game boards. Each board is defined by a file in the **sampleBoards** folder.

You'll want to use these boards as reference when you create your own. Additionally, you will use them to test your program's behavior.

When testing your Battleship solution, use a game board with fixed ship locations (e.g., Classic-1.properties) verses random ship placement (e.g. Classic-Random.properties). This will allow you to modify your algorithm, as needed, and then rerun it against the **same board and ship placement** to compare the results.

Game board details

The board files use a human readable text format (as defined by java.util.Properties) to define the board size and ship placement. You can open the file in DrJava or VS Code since they are plain text files. We encourage you to look at the contents of those files to see how the ships are placed.

For our example, we will use 'Classic-1.properties'. It should look something like this:

```
# Name:
# File Purpose:  A battleship game configuration file.  It defines
# the game board dimensions, ship sizes, and optional specific ship
# placement information.
#
# The commented out game board diagram below is to demonstrate the placement
# that is defined in the variables below. It is not necessary, but is
# useful for visualizing the board you are defining.
#
#        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
#   0 |  |  | *|  |  |  |  |  |  | *|  |  |  |  |  |
#   1 |  |  | *|  |  |  |  |  |  | *|  |  |  |  |  |
#   2 |  |  | *|  |  |  |  |  |  | *|  | *|  |  |  |
#   3 |  |  |  |  |  |  |  |  |  |  |  | *|  |  |  |
#   4 |  |  |  |  |  |  |  |  |  |  |  | *|  |  |  |
#   5 |  |  |  |  |  |  |  |  |  |  |  | *|  |  |  |
#   6 |  |  |  | *| *| *| *|  |  |  |  | *|  |  |  |
#   7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
#   8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
#   9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
#  10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
#  11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
#  12 |  |  |  |  |  |  |  |  |  |  | *| *|  |  |  |
#  13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
```

```
#  14 |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
```

```
# Game board size is always 15x15
```

```
# Comma separated array of ships in length x width format (no spaces).
ships=2x1,3x1,3x1,4x1,5x1
```

```
# WARNING! The values for the following 3 properties must be in the same order as the
# "ships" properties above. If the number of ships or a given ship will not fit on the
# game board, a configuration error will occur upon game start up
```

```
# For example, the first ship in the ships property (2x1)
# corresponds to the first value in the shipsStartPoints, shipsOrientations, and
shipsNames.
# So the 2x1 ship starts at row 12 - column 10, is placed horizontally, and is named PT.
```

```
# Comma separated ship starting location coordinates in row-col format.
# Ship will be placed starting at the coordinate value and
# - if horizontal orientation, ship's length will span columns.
# - if vertical orientation, ship's length will span rows.
shipsStartPoints=12-10,0-2,0-9,6-3,2-11
```

```
# Orientation:  h=horizontal, v=vertical
shipsOrientations=h,v,v,h,v
```

```
# Human-readable names for the ships
shipsNames=PT,submarine,cruiser,battleship,carrier
```

Creating your own

To create your own boards for testing, follow these steps:

1. Open an existing board in your IDE. If you have a board which is close to what you want, use that board.

2. Immediately go to **File > Save As…** and save the file under a new name such as "**MyBoard-1.properties**". It is recommended that you save your board under the "**sampleBoards/**" folder.

3. Fill in your name on the first line # Name:

4. Modify the ships on the board as you desire. To do this, you only need to modify two values – shipsStartPoints and shipsOrientations. Explanations for each field are included in the comments above them. **Leave all other fields unchanged.**

5. Save the file! Now, run your TemplatePlayer with the 'runWithBoardPrompt' method and select your new custom board. The game should start playing if you configured your board correctly.