Battleship 2021 Setup, Submission, Reference Guide

Table of Contents

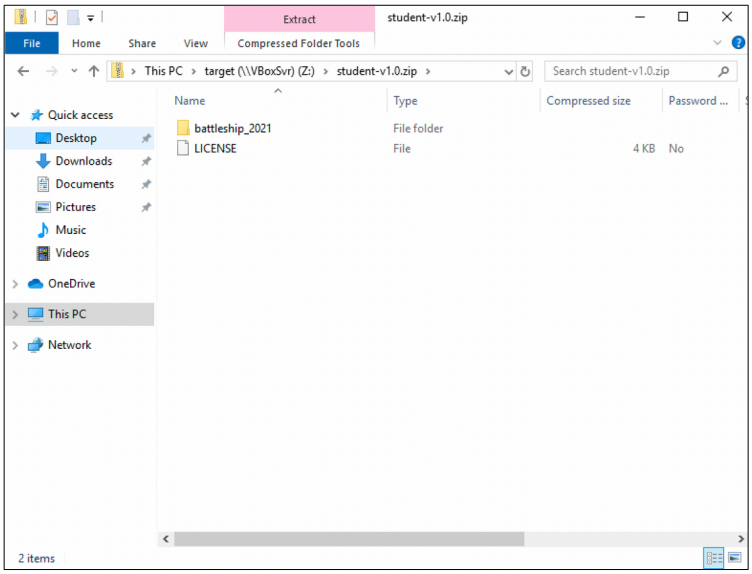
Setup

Download

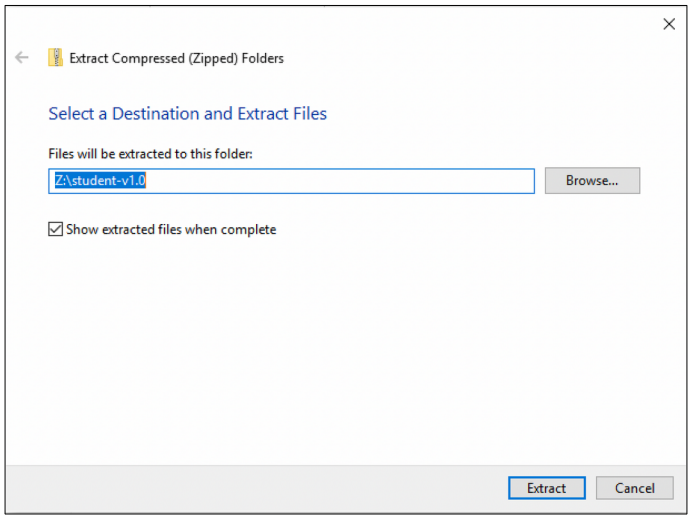
1. Obtain **student-v1.0.zip** from your teacher
    a. Save it to a location like the Desktop or (preferably) to your personal drive (e.g. X: drive).

Extract Files

1. Once the ZIP file is downloaded, a window (shown below) may open showing the contents of the ZIP.
    a. If no window pops up, find and double-click on **student-v1.0.zip** to open it.
    b. The window should show **battleship_2021** and **LICENSE** inside the .zip.



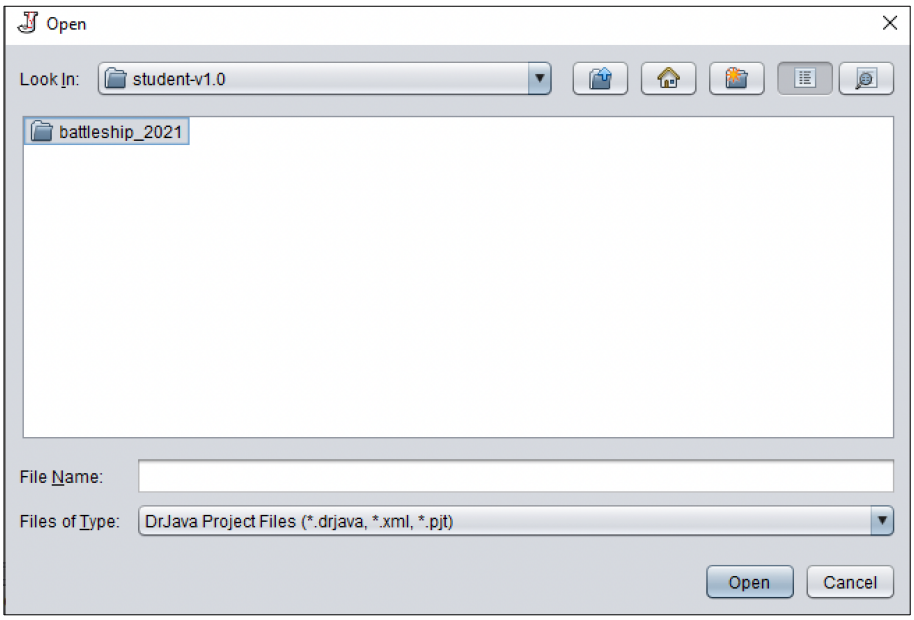
2. In the window, click ‘Compressed Folder Tools’ and click ‘Extract all’
3. A dialog box (shown below) will appear asking where to extract the files.



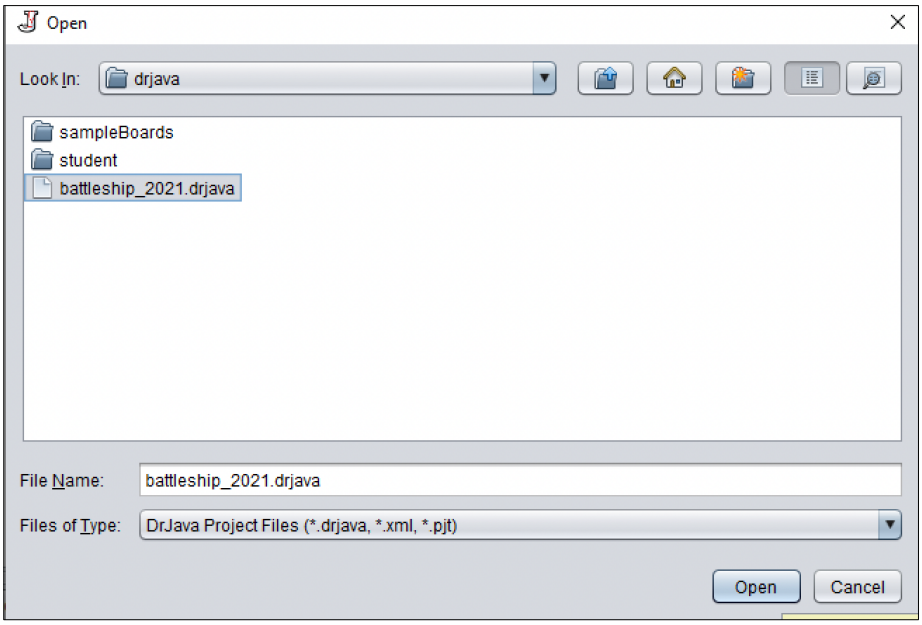
4. Click ‘Extract’. The folder and file should now be extracted to the listed location. **Remember this location!**

DrJava

1. Open DrJava
2. Under Project, click Open… to get the following dialog box



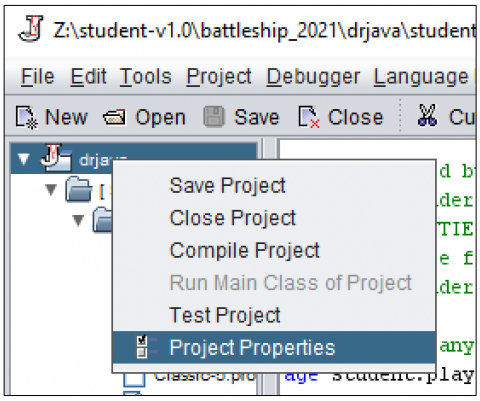
3. Navigate to your remembered location where you extracted the files
4. Enter the **battleship_2021** folder and then the **drjava** folder
5. Select the **battleship_2021.drjava** file and click Open



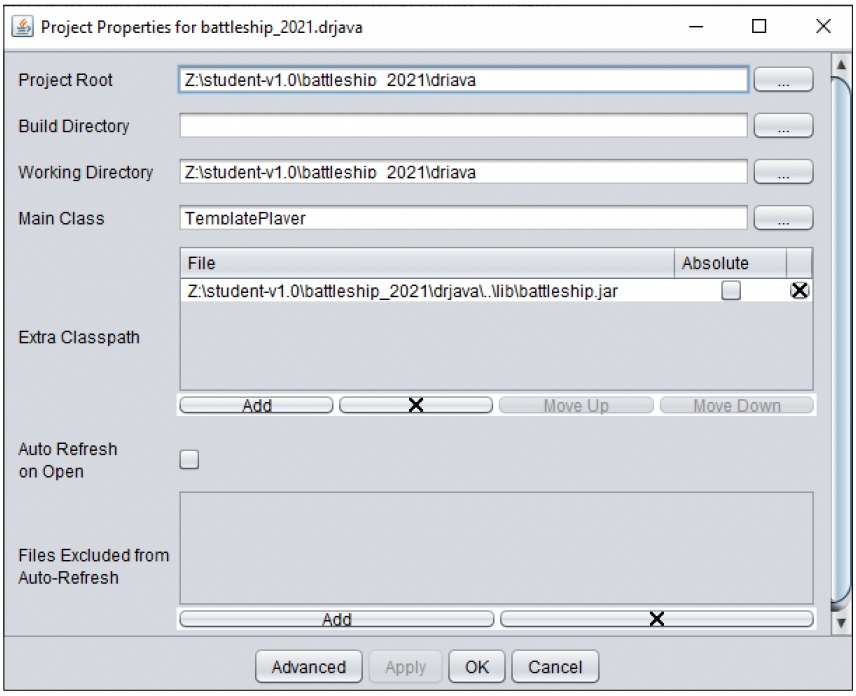
6. The Battleship project should now be in your DrJava workspace with the **TemplatePlayer** ready for you to code. However, before you start, let's ensure a couple of things.

Ensure Project is Configured Correctly

1. In your DrJava workspace, right click on the new project and select 'Project Properties'



2. A dialog box should appear with several items.
   a. Ensure the 'Main Class' property is set to **TemplatePlayer**
   b. Ensure the 'Extra Classpath' property is set to **`<YourLocation>\battleship_2021\drjava\..\lib\battleship.jar`** AND 'Absolute' is not checked
   c. <u>If these mentioned items aren't correct, you won't be able to run your program!</u>



3. Click Ok to exit the dialog box

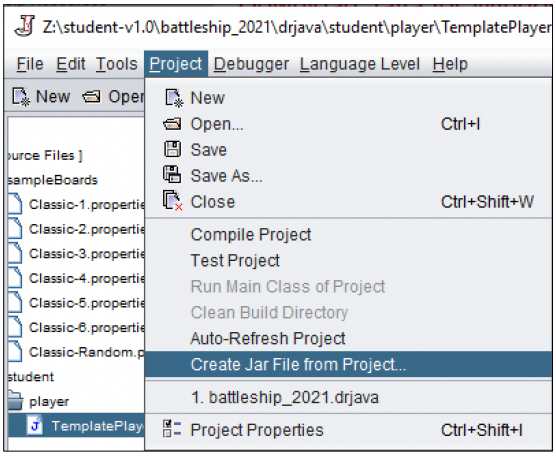
Compile and Run

1. Compile the battleship project by clicking 'Compile Project'
   a. **You <u>should</u> encounter an 'illegal start of expression' error.** Enter your name as a String and recompile
2. Click 'Run' once your project is compiled to test that the application works
3. Happy coding!!!!

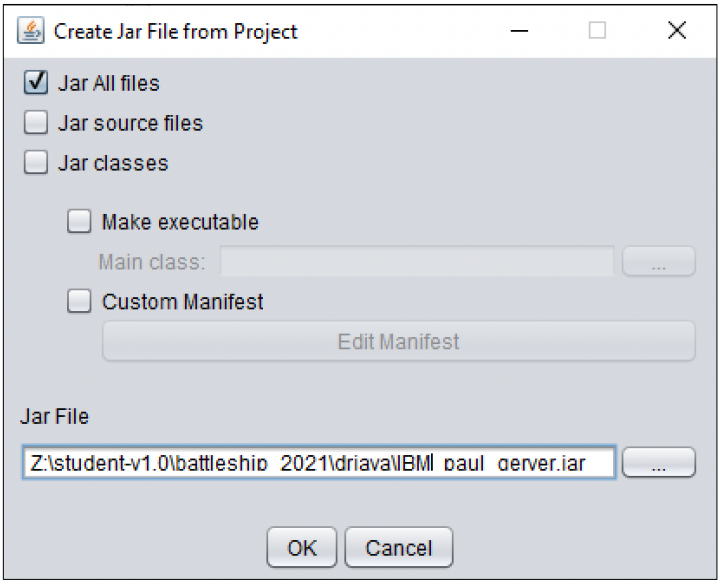
Submission

Create a Jar

1. In DrJava under Project, click 'Create Jar File from Project…'



2. A dialog box (shown below) will open. Ensure 'Jar All Files' is selected!



3. Edit the 'Jar File' line to be **<path>\battleship_2021\drjava\schoolname_first_lastname.jar**
   a. Ensure you use the underscore '_' between fields
4. Click OK
   a. A window should pop up saying the jar creation was successful. Click OK
5. The jar should be located at the **<YourLocation>\battleship_2021\drjava**\ folder
6. Submit **this** JAR file to teacher, as instructed by them!

Reference

battleship_2021 Contents

- *doc* – All helpful documentation about and for the Battleship competition. Be sure to read these or you'll hurt ol' Paul's feelings.
    - *javadoc* – Has several HTML files about the classes used in Battleship. Don't be alarmed! See the Javadoc reference for more info.
- *drjava* – The DrJava project and the Player that you get to implement.
    - *sampleBoards* – Contains battleship boards (enemy ship placements) that your Player plays against. Use these to hone your algorithms skills or focus your algorithm on a single board type for mastery.
- *lib* – Has the **battleship.jar** which contains all the compiled code to play Battleship. It really does make the world go round.
- *misc*
    - *src* – Houses all source code used for the Battleship game

Javadoc

To view the javadoc, you will need to go to the **battleship_2021\doc\javadoc** folder and open **index.html**. The documentation should open in a web browser (Firefox, Internet Explorer). The web page allows users to view all interfaces and classes in Battleship.

If you have any questions about an interface, class, or method, the Javadoc should be your first stop. For example, the Javadoc for the initial `TemplatePlayer` can be found here and links to other classes that it uses or might use.

Running the Game

See the main method in the `TemplatePlayer` for instructions and code on running the game. Main invokes `PlatformImpl.onePlayerQuickStart` to get you running quickly. If using the default quick start, a file dialog box will open in the **sampleBoards** folder where you can select a board to play against. Once selected, the battleship GUI will appear and the console will show a summary of the results of the game.

You may also invoke the main method `my.battleship.PlatformImpl` to have more control over the behavior of the system. See the javadoc or source code for `my.battleship.PlatformImpl` for details.

Game boards

student-v1.0.zip supplies an initial set of game boards. Each board is defined by a file in the **sampleBoards** directory. For the game, use the boards whose name starts with 'Classic-'.

The board files use a human readable text format (as defined by java.util.Properties) to define the board size and ship placement. Think about how your player algorithm will deal with the different kinds of boards. You can open the file in Dr Java since they are plain text file. We encourage you to look at the contents of those files to see how the ships are placed.

To test your program's behavior, you will want to 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** and then compare the results. The competition will use a mix of fixed and random board placement. You do NOT need to create your own board for this project, but you might not be able to test your player implementation sufficiently with only the boards provided.

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

1. Open an existing board in your favorite text editor. If you have a board which is close to what you want, use that board. For our example, we will use 'Classic-1.properties'. It should look something like this:

```
# File Purpose:  A battleship game configuration file.  It defines
# the game board dimensions, ship sizes, and optional specific ship
# placement information.
#
# Note that if the number of ships or a given ship will not fit on the
game
# board, a game start up configuration error will occur

#      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 in row x column format (no spaces).
gameBoardRowsCols=15x15

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

# NOTE!! The values for the following 3 properties must be in the same
order as the
# "ships" properties above.

# The following two properties are required for placing ships at
specific
# locations on the game board.

# 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
```

2. Immediately perform a **Save As** operation and save the file under a new name such as "**MyBoards-1.properties**". It is recommended that you save your board in "**sampleBoards/**" directory.
3. 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.
4. Pass the name of the new file on the `PlatformImpl.onePlayerQuickStart()` call in the `main` method. For example, the unmodified `main` method of "TemplatePlayer.java" file looks like the following on line 81:

```
public static void main(String[] args) {
        System.out.println("In main...");

        // The following allows you to run the game from...
        // method versus running from the command line.  To ...
        // must be included in your environment's classpath.
        PlatformImpl.onePlayerQuickStart(
                // The name of the battleship game board...
                // TODO Change as desired. This board has...
                "sampleBoards/Classic-1.properties",
                // The Class object representing your player...
                // game uses this to plug your code into the...
                TemplatePlayer.class,
                // true to show the GUI. This optimizes the...
                true);
```

Now, modify the call to include your filename (e.g. "sampleBoards/MyBoards-1.properties") instead of "sampleBoards/Classic-1.properties". DrJava will then find and load your board file.

IBM-Supplied Source Code

The class **student.player.TemplatePlayer** is provided as a skeleton for you to create your implementation. It uses a very naïve algorithm. You may copy it and make the modifications as indicated by the TODOs in the code.

The IBM volunteers provide javadoc and source code for the battleship game code, sample player implementations, sample game boards, and class diagrams. Areas of interest include:

- "my.battleship" is the main Java package that contains the interfaces and classes the student program will use. The **battleship.jar** will contain the executable classes and properties files.
    - **my.battleship.PlatformImpl** is the main control class that orchestrates the players and game(s).

Class Diagram