

Assessment 4 Readme

This directory contains the source code for Team BHD's Assessment 4 submission. It is based on the Assessment 3 project inherited from [Team PSA](#), which was in turn based on [Team WAW's](#) Assessment 2 project.

Building and Running the Game

Using Ant

The easiest way to build and run the game is to use the [Apache Ant](#) Java build system. It can be downloaded from the website linked above, or installed through a package manager, such as apt-get on Ubuntu, [brew](#) on Mac, or [cinst](#) on Windows.

Once Ant is installed the game can be launched by opening a terminal window in the project directory and then executing `ant`, which will automatically build and run the game (to just build without launching run `ant build`).

The included unit tests can be run with `ant test`. Running `ant report` will generate a html version of the test report within the junit directory.

The `ant javadoc` command will generate JavaDoc documentation within the doc directory.

The `ant clean` command will clean-up the working directory, by removing both the generated binaries, JUnit reports, and JavaDoc documentation.

Using Eclipse

The following instructions assume that Eclipse is already installed and configured on your computer. If not it can be downloaded by following [instructions on the project website](#).

Firstly, we'll need to import the downloaded project into Eclipse. In the main menu go to *File > Import...* and in the dialog box that appears expand the "General" category and double-click "Existing Projects into Workspace". On the next screen click the *Browse...* button next to "Select root directory" and find the directory containing this readme file. Then leave all the checkboxes in the bottom half of the dialogue unticked and click *Finish*.

To run the game, select the top-level "Game" item in the Package Explorer on the left-hand side of the screen and then go to *Run > Run Configurations...* In the window that appears, expand the "Java Application" item in the left-hand column and then double click on the "game" item beneath it.

To run the tests go to the same dialog box and double-click the "tests" item within the "JUnit" category.

Packaging for Distribution

This section describes the steps needed to produce standalone executables (containing the game and all its dependencies) suitable for placing on the team website and for including in the assessment submission.

The first step in the process is to create a runnable JAR file, which will contain the game and all platform-independent dependencies. The easiest way to do this is to run `ant mkrunnable` within the project directory. That command will create the file `runnable.jar` within the `dist` folder.

The next step is to use JarSplice (included in the tools directory) to create a so-called "fat JAR" which includes the needed native libraries. In the JarSplice screen, add the `runnable.jar` file on the "ADD JARS" tab, add the entire contents of `lib/natives` on the "ADD NATIVES" tab and enter "stateContainer.Game" in the "MAIN CLASS" tab. Finally click *Create Fat Jar* on the "CREATE FAT JAR" tab and choose an appropriate location. The JarSplice program can also be used to create .exe files for Windows and .app files for Mac OS X.

Style Rules

We're using the `astyle` program to automatically enforce some of our style rules, for instance that 4 spaces should be used for indentation and that spaces should be left around operators. Before committing code into git you should run `astyle --options=astyle.conf src/*.java` in the directory containing this readme file. The comments in the `astyle.conf` file give full details of the precise rules we're trying to stick to.