

Assignment 4

COMP SCI 2ME3, SFWR ENG 2AA4

March 20, 2017

The purpose of this assignment is to design and specify a module (or modules) to store the state of a game of Battleship. The modules cover the Model portion of the Model View Controller design pattern. The rules for Battleship can be found at the following web-page:

<https://www.thespruce.com/how-to-play-battleship-411069>

Your assignment is to write a module that stores the state of the game board and the status of the game. You do not need to worry about modules that display graphics, or control the game play, or determine the strategies of a computer opponent, etc.

Bonus marks are available for students that both implement and test their specification.

Deadlines

- Specification: due 11:59 pm Apr 3
- Code and Test Suite (Bonus): due 11:59 pm Apr 3

Step 1

Submit a report showing your design specification of the MIS for the game state module. If your specification requires additional modules, you should include their MISes as well. It is up to you to determine your modules interface; that is, you decide on the exported constants, access programs, exceptions etc. You also determine your state variables and specify the semantics for your access program calls. When working on your design, you may want to consider the following points:

- You probably want a game state for each of the two players. This would make the module an abstract data type, not an abstract object.
- The state of the game board with the battleships could be modelled as a two dimensional sequence.
- A separate board could be kept for each player to track the shots they have taken.
- You will need a routine to initialize the board.
- You will need to be able to tell if the placement of a ship in the initial configuration is valid or not.
- You will need to be able to determine whether a shot is valid or not.
- You will need to be able to inspect the state of any cell of the game board.
- You will need to be able to tell when a ship has been sunk.
- You need to be able to tell when the game is over.

Bonus Step 2

Submit Java or Python code that matches the specification given in the previous step. You should also submit code that tests your module(s) using PyUnit or JUnit. You are free to document your source code as you see fit. That is, you can use doxygen, javadoc or pydoc, or you can use regular code comments. Your code should include a makefile, with rules `make test` and `make doc` (if appropriate).

Your submission of code and test cases should be to our git repo.

Notes

1. Your git repo is organized with the following directories at the top level: **A1**, **A2**, **A3**, and **A4**.
2. Inside the **A4** folder you will start with initial stubs of the files and folders that you need to use. Please do not change the names or locations of any of these files or folders. The structure of your project files and folders should look like this:
 - A4
 - * Makefile
 - report

- * report.tex
- * report.pdf
- src

3. Please put your name and macid at the top of each of your source files.
4. Your program must work in the ITB labs on mills.
5. **Any changes to the assignment specification will be announced in class. It is your responsibility to be aware of these changes. Please monitor all pushes to the course git repo.**