**CSCE 155N Matlab Programming Project 2 – Summer 2013**

**Assigned: Tuesday 7/29/2013**

**DUE DATE:**

**Tuesday 8/6/2013 at 11:59 PM**

**(hardcopy in class on Wednesday)**

**Battleship!**

**Problem Statement:**

See the web site <http://www.learn4good.com/games/board/battleship.htm> for an online version of the popular board game Battleship. This project has you implementing the game in Matlab. However, you do NOT need to replicate the animation, graphics, GUI environment, or sound effects!

The site has a player compete with the computer which randomly places its ships and fires at random locations until it hits something, at which point it searches for the rest of that ship. The player selects a name for his persona, then places his ships on a tract of ocean which the computer obligingly does not “see” to augment its play. Your implementation may perform similarly or, if you wish, pit two human players against each other, in which case attention must be given to preventing the players from seeing each other’s placements! Do check <http://www.datagenetics.com/blog/december32011/> for extensive analysis of strategies for winning which you MIGHT wish to implement for the computer to use (or for your own use!) Rules can be found at <http://boardgames.about.com/od/battleship/a/Rules-of-Battleship.htm>.

The size of the ocean grid and the number, types, and sizes of your ships may be the same as the online version, or you may have the players select from various options. Similarly you are welcome to consider variations to the rules.

You are to work in teams of two or three students to design prototypes in Matlab. Larger teams may be allowed by permission, but there would be higher expectations. You may keep the same teams as for the first project, but you are also welcome to make changes.

**Collaboration:**

Work together as a class on any or all aspects of the research and design. Ideally take advantage of the talents of each member of the team, but recognize that each is responsible for the entire project! This means being prepared to answer questions on the code even if your allocated task was to write the report. It is essential to keep track of who did what and where any useful information was found. Record each time you helped someone else and each time someone helped you. Keeping a log is highly recommended. Note that Piazza keeps a record automatically!

**What and How to Submit:**

Read and have your program conform to the “Program Documentation Guidelines” which are online.

By the deadline hand in electronically the two files, battleship.m (the Matlab script file for the game), and battleship.doc (which contains summaries, documentation, and sample runs). Only one copy per group needs to be submitted – it does not matter who does the handing it. In class the day after the deadline, hand in the hardcopy version, stapled together with the cover page in front.

Each team member should electronically on his/her own account submit his/her own analysis of the relative contributions of all the members toward the project. This is in addition to the acknowledgement section of the main report. Assuming allocation is fairly even, all will receive the same grade.

The Word document should contain the following, all carefully labeled:

* Cover page with name(s) and the account under which it is submitted, title, date submitted, etc.
* A discussion of the features you implemented in the project. Describe how they work and what Matlab options were used to program them. This should be at a fairly high level, not a line-by-line analysis of the code.
* An “instruction manual” that a non-programmer can use to run the game.
* An annotated cut and paste sample dialog of the running of the program. (Hint: Use the ‘diary’ command and/or capture screen shots.)
* A discussion of the testing that was performed. This should include testing of each component as it was being built, and testing of the final program ensuring that it works properly under a comprehensive range of conditions.
* An annotated cut and paste of a sample dialog, demonstrating how your program responds to extreme and faulty input. (This could be combined with the previous section.)
* Acknowledge all collaborations (both internal to the team and external), detailing what each person contributed individually, and what was done jointly. Indicate approximate percentages of the work contributed by each person in design, coding, testing, documentation, and report preparation.

**Grading Criteria: (subject to some modification)**

* Program functions as intended – 30%
* Program logic is well designed – 20%
* Documentation guidelines are followed – 20%
* Handin Documents formatted and arranged as specified – 10%
* Testing is comprehensive – 10%
* Quality of the user’s manual – 10%