Lab 5: Battleship!

Due Date
11:59 PM, Wednesday November 4\textsuperscript{th} 11:59 PM
Email zipfile “LastName-Lab5.zip” to cse436ta@gmail.com

Assignment – 20 points with up to 4 points of extra credit
In this assignment, you will practice all of the skills you have learned in the class so far to design a game of Battleship, complete with touch events and more. You will also have the opportunity to go above and beyond what's required, in order to earn extra credit points. As in Tic-Tac-Toe, you will be asked to design both a one and two player game.

We suggest that you create an iPad application for the increased screen size.

Begin by reading about the game: \url{http://en.wikipedia.org/wiki/Battleship_(game)}

Note that much of the implementation detail has been left up to you.

The Basics
The two players should each have two copies of their own ten by ten grids. The first grid (#1) should be used to record where the places his or her ships. The second grid (#2) should be used to record whether a player hits or misses an enemy ship.

Each player will begin by placing five ships on their own grid #1. These ships may be placed horizontally or vertically on the game board, such that no two ships overlap, and each ship must fit entirely on the grid.

When the game starts, you should display each of the five ships graphically. There are four different size ships: one of length five, one of length four, two of length three, and one of length two. The human players must be able to “drag and drop” a ship onto the specified position in the grid using touch events. If a computer player is playing, then its ships should be placed randomly on the grid, and must not violate any of the previously specified rules for ship placement.

Furthermore, no player should be able to see the ships or grid of the opposing player. If two humans are playing, the game should decide which grid to show, depending on whose turn it is to move.
Once each player has placed the five ships, you should randomly decide which player starts the game. Each player will then click on a grid location on grid #2. If the enemy has a ship in that location, you should display an X on the location on grid #2. If the enemy does not have a ship in that location, you should display an O. After each move, the next player gets to move.

If at any point a ship has been sunk, there should be some clear graphical indication. Furthermore, you are asked to play a short sound clip or special effect sound of your choosing (keep it appropriate). You should consult with the iOS and Xcode documentation on how to do this.

If a computer is playing, and has no partial ship hits, it should randomly target a position on the grid. If the computer successfully hits a ship, it should try and sink the ship. That is, it should logically infer possible positions for the rest of the ship, and continue to place hits in these positions. Remember that ships are continuous and can only be placed horizontally or vertically. After successfully sinking a ship, the computer should go back to randomly selected a position.

The game ends when a player has successfully sunk all five opposing ships. When this occurs, you should play some victory music (which should be different to the ship sinking sound effects).

Feel free to implement anything that you think will make the game look better. For example, you might include a statistics section showing the number of games won and lost (using NSUserDefaults for example). You might include the current score at the top of the screen showing the number of ships sunk by each player. Going above and beyond will earn you extra credit points.

You should also prepare a write up detailing your design choices, how you went about approaching the design process, what software engineering ideas you made use of, as well as any thing else you think you should mention about your game or the implementation. Please format it as a PDF document, and include it in your zip file for the lab.

You are encouraged to explore additional features of iOS that have not been discussed in class. For example, use Game Center to add achievements, or allow for multiplayer games using two devices.

**Grading**

The usual grading deductions apply (for turning in your lab late or for having warnings and/or errors). Note that the late penalty is 10% per day, so for this lab it is two points. If your lab is turned in late, you will not be eligible for any extra credit.
For this lab you will be graded on:

2 Points – Write up, approximately one page. You should include information about your game, and the design process. Please include it as a pdf document in a zip file. Please use formal English, with complete sentences and paragraphs.

6 Points – Initialization of the game. The bulk of this will include using touch controls to place ships on the grid. Your game should also ask for one or two players, and look professional.

4 Points – The grid is displayed correctly at all times. Under two player mode, grids alternate each turn. Missed locations and correct hits should be displayed, and up to date.

4 Points – The game continues until one player has successfully sunk all of the opponent’s ships. There should be some indication of who has won.

4 Points – Sound effects are played when a ship is sunk.

You also have the opportunity to earn up to 4 points of extra credit. At most 2 points will be based on making your game look professional. To receive the full amount of extra credit points, you must add a sizable feature such as Game Center. You are free to choose this feature on your own, or consult with the Professor or Teaching Assistants if you need help.

**Note:** You cannot earn extra credit if you fail to attempt one of the required pieces. i.e. you cannot skip the sound effect requirement, and add in your own feature. Your game must have working touch controls, sound effects, correct grid displays, and correct win detection in order to earn extra credit points.