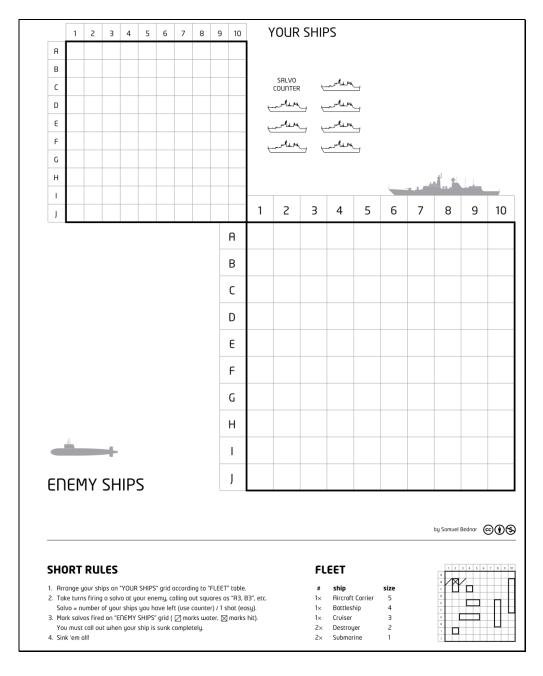# McMaster University, CS 2ME3 / SE 2AA4: Jan – Apr 2013

Assignment 3 (& 4). Due Wed 3 April 2013, 11:59 pm. Submit on Avenue – all files in one zip file.

This assignment is to build a computer version of a popular two-player game – Battleships.

The following is an example of the paper/pen version of the game. Each player positions their fleet of ships on the primary grid and keeps their location secret. Once play starts, each player has a turn to fire at the enemy ships. This is done by specifying a grid location – B4, E9, etc. If a ship is located at that location, the owner of the ship indicates a "hit". Otherwise the owner indicates a miss. Each salvo fired is recorded on the grid as a "hit" or "miss". If all the squares on a ship have been hit, the owner indicates that the ship has been sunk. The first player to sink all the other player's ships wins the game. The example shows the number of ships in a fleet and the size of each ship (in squares).

by Samuel Bednar

SHORT RULES

1. Arrange your ships on "YOUR SHIPS" grid according to "FLEET" table.
2. Take turns firing a salvo at your enemy, calling out squares as "A3, B3", etc.
   Salvo = number of your ships you have left (use counter) / 1 shot (easy).
3. Mark salvos fired on "ENEMY SHIPS" grid (☑ marks water, ☒ marks hit).
   You must call out when your ship is sunk completely.
4. Sink 'em all!

FLEET

#	ship	size
1×	Aircraft Carrier	5
1×	Battleship	4
1×	Cruiser	3
2×	Destroyer	2
2×	Submarine	1

McMaster University, CS 2ME3 / SE 2AA4: Jan – Apr 2013

Design, implement and test a computerized version of Battleships in which the player plays against the computer. The game must use a graphical user interface, and the appropriate grids must be displayed on the screen. Further requirements are as follows:

1. Document the design and tests. You must include some version of a module guide (class guide), and an MIS and MID for each class. The design is clearly of primary importance – this is a software design course ☺
2. Document usage of the game and include appropriate screen shots taken of the implemented game.
3. Do not spend too much effort (initially at least) on the strategy the computer uses to play against the human player. However, there must be more strategy than simply firing a salvo generated by random numbers.
4. The player must be able to stop and resume a game.
5. The game shall record appropriate statistics. The player can be anonymous (no statistics) or can enter a name. Statistics are then accumulated for specific players. Best score for each player and overall shall be stored. It shall be possible to clear statistics for a specific player, and/or the best score overall (and the name of the player who achieved it).

Submit the entire project by zipping all relevant files. Make sure that there is an executable version of your game included. If you use an IDE, such as Visual Studio, extract the .exe file and submit it in the zip file (as well as the IDE project file). Indicate very clearly what computer platform(s) your game runs on.

[50 marks]