# **COSC 111 – Assignment 9**

# **Battleship – Part 2**

**NOTE: This assignment does not use your code from Part 1! Instead it builds of code that is provided for you.**

For the second part we are going to make an Artificial Intelligence (AI) bot to play battleship for us. We are also upgrading to a full size, 10 x 10 game board.

We will have a class competition later, putting your AI bot against other students AI bot. The method you write for your AI guesser needs to be formatted carefully so that it will work in the class competition. Your AI also cannot have access to the actual ship placements, as it will not have that in the competition.

I recommend that you start with the most basic AI which is to find the next available spot that hasn’t been guessed, and guess there. Then finish the rest of the assignment before trying to improve your AI.

Your AI method declaration must look like this:

public static String makeGuess(char[][] guesses){

The 2D array guesses is a char array of the results of your guesses so far. Each char in the **guesses** array has a particular meaning:

‘.’ – no guess yet

‘O’ – miss

‘X’ – hit

‘1’ – Patrol Boat, has length 2, and has been sunk

‘2’ – Submarine, has length 3, and has been sunk

‘3’ – Destroyer, has length 3, and has been sunk

‘4’ – Battleship, has length 4, and has been sunk

‘5’ – Aircraft carrier, has length 5, and has been sunk

**Bonus marks (2% of final grade): You must get your average number of guesses to complete a game under 50. I recommend you check out this site for a good strategy:**

[**http://datagenetics.com/blog/december32011/index.html**](http://datagenetics.com/blog/december32011/index.html)

# **Starter Code**

I have posted 3 files to get you started:

BattleShipTools.java – There is no need to modify this file at all. It is a collection of methods that make testing your code a bit easier. You can use it to generate a random board, or use a set board. It also has a printBoard method.

KenRandom.java – This is a terrible AI that just randomly guesses. You AI should be way better, but you can use it to check things out to start.

RunGame.java – This file is an example of how to use the BattleShipTools file to run a game. You don’t need to modify much this file to start, as it is setup to call the AI (currently set to KenRandom) for a single game. Eventually you will most likely want to change this file to run it on many random boards, so you can determine your average.

# How good does it need to be?

Your AI, for full credit, must win on average in under 65 moves. I will run your code on 10,000 random games, and compute the average. You can achieve this by having 2 modes, hunt and target. In hunt mode, you don’t know where any non-sunk ships are. In target mode, you have hit a ship, but it hasn’t been sunk yet. Once you have this setup, you are well on your way to that huge bonus!