

Week 4 Report

David Dudson

This week I have tidied up some issues surrounding the build scripts, Also I created the branch SoGaCoGameState and fully ported our battleships game into the SoGaCo framework. This way Aidan can help out Mitchell with blockly. I wrote all the code from scratch using the highly efficient Java8 Streams for array processing. Aidan's previous work on the prototype of the battleships game has been used as a reference and the algorithm for generating the ships was a rewrite based directly off his implementation, a lot of the other parts had to be rewritten due to the nature of the SoGaCo framework and implementation details. The new gamestate technically could have an infinite number of bots play against each other if SoGaCo could support it. The Ships and grid size is fully changeable so we could quite happily have a 5,000x5,000 grid with 1,000 ships and 100 bots if we wanted.

The random seed:

I've made sure the random generator, for ship placement is static, or can be saved. This means in the future if we want to have a specific set of 100 gameboards to test a bot on it is quite possible to do that, or a debug gameboard that never changes, we can.

The changes to the game structure:

- 4 states: Unknown, Miss, Hit, Sunk, this way a bot can determine whether or not it needs to keep attacking around the same area (This is different advantageous to bots vs the real game by milson bradley)
- If you get a hit, take another turn(Although the usage of this rule wasn't unanimous, we decided that it makes the game a lot more interesting, especially when detecting the good bots from the bad.)

Things to work on for next few weeks (All are possibilities, not necessarily set in stone, basically my backlog for scrum):

Implement tests: To try and get things done for the milestone, i avoided test coverage. unfortunately since I couldn't get everything working in time, we have been delayed anyway. Although I have designed the code to be fairly abstract and easy to test.

- Add guava predicates: Most of my functions currently aren't guarded as well as they could be, I want to implement predicates now guava is a dependency.
- Remove apache commons, once we switch the UI to the SoGaCo backend i will remove apache commons as my entire version of battleships doesn't use it.

- Find a solution to the current heroku autodeploy problem: at present we are manually deploying our project to heroku due to some dependency issues with heroku not understanding.
- Refactor the codebase(Renaming): During the course of making the gamestate, my naming has been a bit off. I need to go back through and tidy some things up.
- Refactor the codebase(Structure): From a high level design point of view some things seem a little out of place, I want to tidy these up. Eg How the ShipMap and BotMap are both subtypes of gameboard but shipmap is where the gameboards are created.
- Tidy up loose ends with the API and what coders can actually access.
- Add a ranking system for bots. Based on 100 set games, what was the average score.
- Allow a debug gameboard