Final Project Report

Author: Philip Gilbreth

For my project, I created a clone of the popular game Battleship. I won't go too much into how the code actually works on here. I have a fleshed out Javadoc attached with this project if you want to check that out. Let me run through exactly how I met the requirements:

1. Object Oriented elements
    a. Classes – I use several, including a couple of classes I generate multiple Objects out of, like Space and Ship
    b. Subclasses – PlayerGrid and PCGrid are subclasses of the abstract class Grid. PlayerGrid and PCGrid had enough elements in common to stem from the same parent classes, while having enough different to constitute their own classes. I never need a pure Grid object, so I leave it, and the different methods to be inherited, abstract.
    c. Abstract/Interface – See above
2. Code Elements you Utilize
    a. Collection Classes – I use ArrayLists throughout my project. I maintain the logical board (not the display board) using a mult-dimensional ArrayList of spaces, which allows quick querying of the spaces. I also use an ArrayList to quickly reference each player's ships
    b. Exception Handling – I use Exception Handling at several points, especially when it gets risky that I might run off the end of an array and get a NullPointerException. Definitely made debugging easier (See line 114 of Grid.java). I also use a try catch block when the application pauses when the PC is taking its turn (line 266 of BattleshipController).
3. Clearly Defined Model – attached as ModelUML.png
4. Multiple states -- My application has two main states, the set-up state and the play-game state. To switch between the two states, listeners are installed/uninstalled, shapes are moved about so they are in the right place, and colors of the grids change. Play the game to see these states.
5. About Information – Press the About button at any point while playing the game

Hopefully this assists you in your grading. This project took a lot more work than I originally anticipated, and I am glad it's finished. I think it looks nice though, and it works pretty well. It still has a few bugs that I didn't have time to iron out, and the AI is not very intelligent, but it's a lot better than other things I have made so far. Let me know if you have any questions or concerns!