OOAD Project 6 Report
Battleship

Jake Luoma
Abhinivesh Palusa
Nikhil Prabhu

Statement on Final State of the System

Features Implemented

- User login
- User logout
- Display user statistics
- Change display settings
- Generate a new game
- Game autosaves
- Load a game
- Ship placement
- Take turn (attack)

We implemented a fully functional, command line-based, standard Battleship game. A player can log in, save/load games, track their lifetime performance, and play against a reasonably difficult AI.

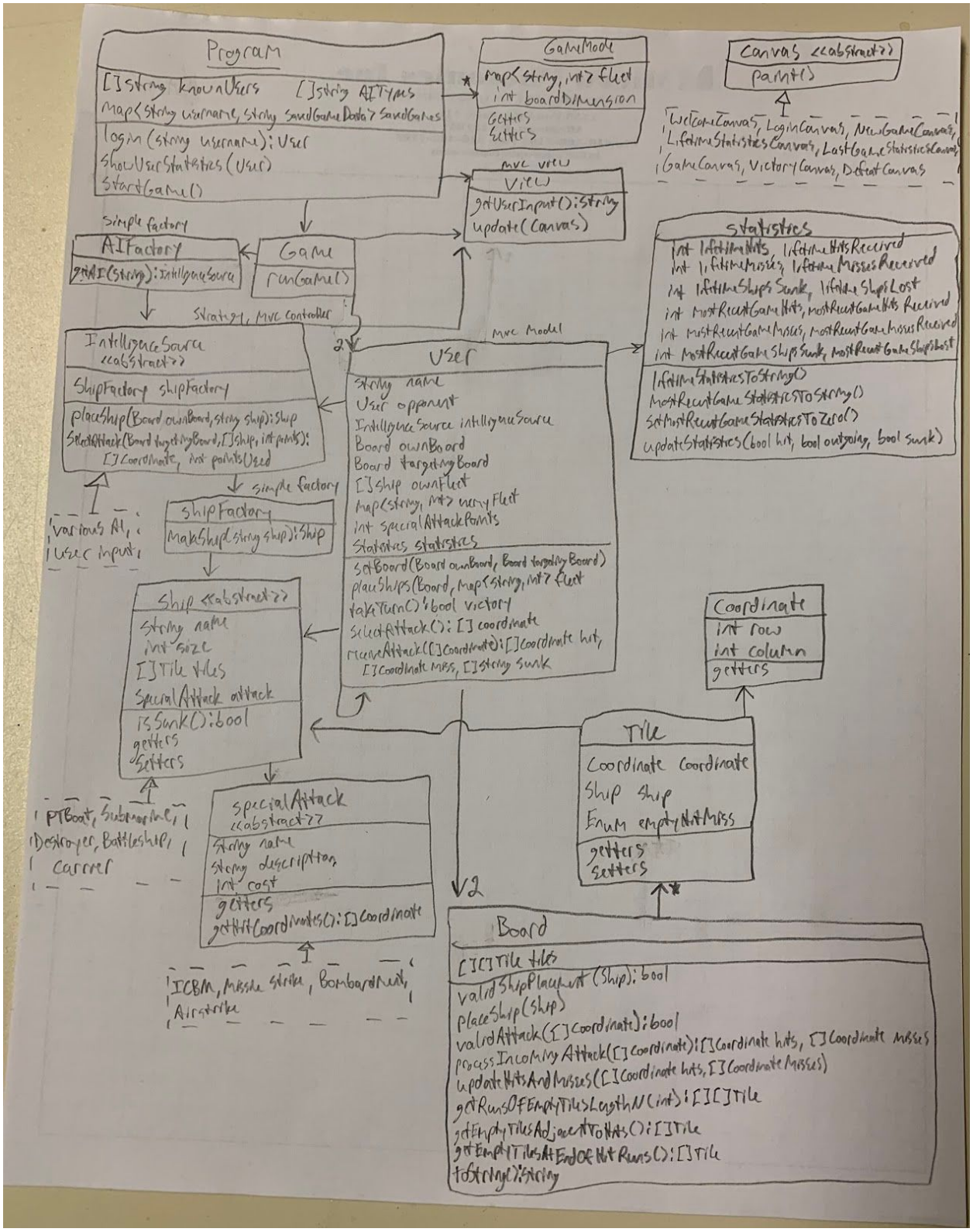
Features Considered but not Implemented

- Special attacks
- Multiple AI difficulties
- User-selected board size and number/type of ships

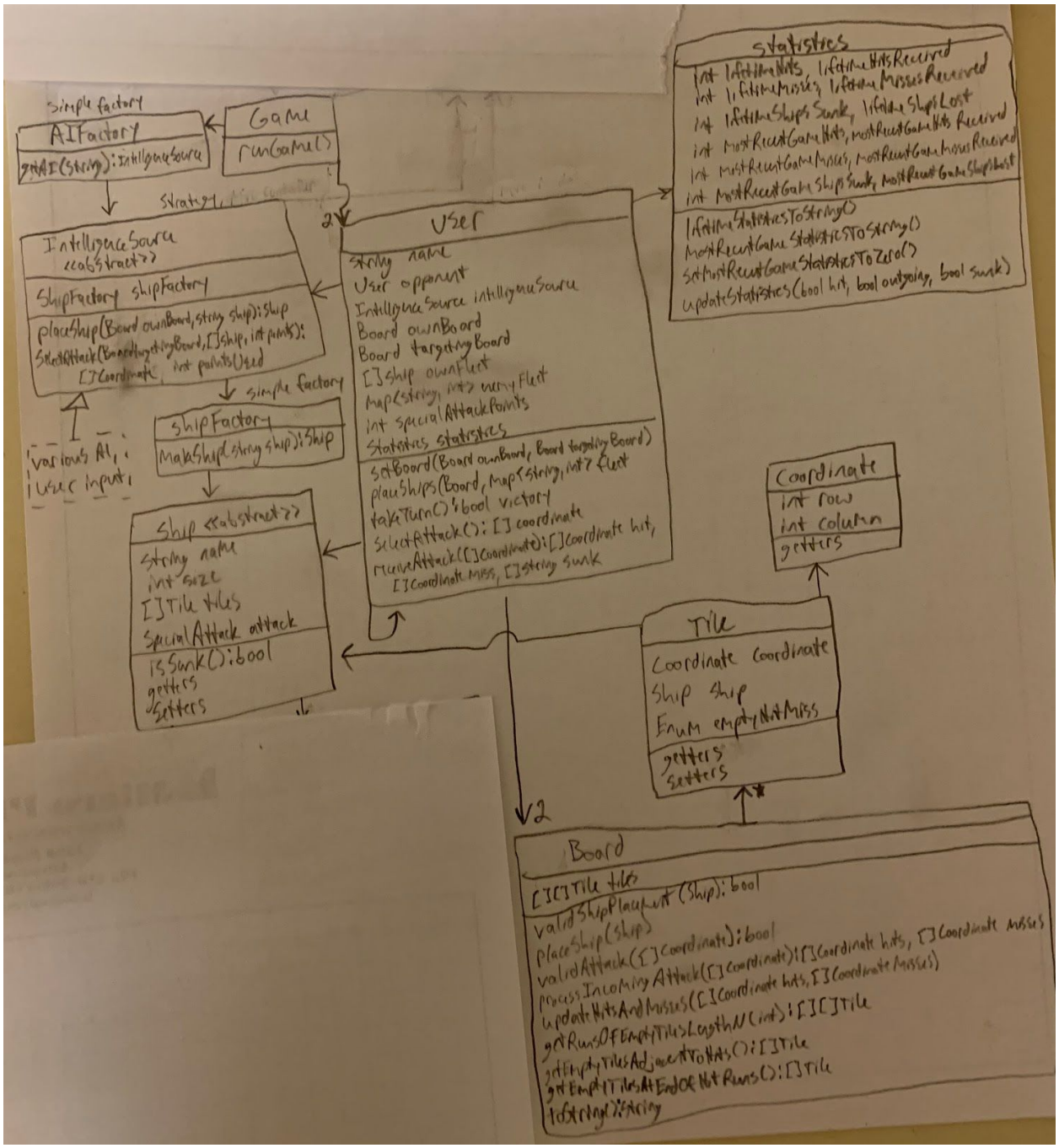
We originally considered adding special attacks and user-selected game modes but chose instead to stick to the standard game since that turned out to be a significant amount of work in and of itself. Similarly, we considered adding multiple AI difficulties but chose to only do one since an easy (meaning random) AI is not fun to play against and the effort to make a very hard AI was instead put into the other features.

Final Class Diagram and Comparison Statement

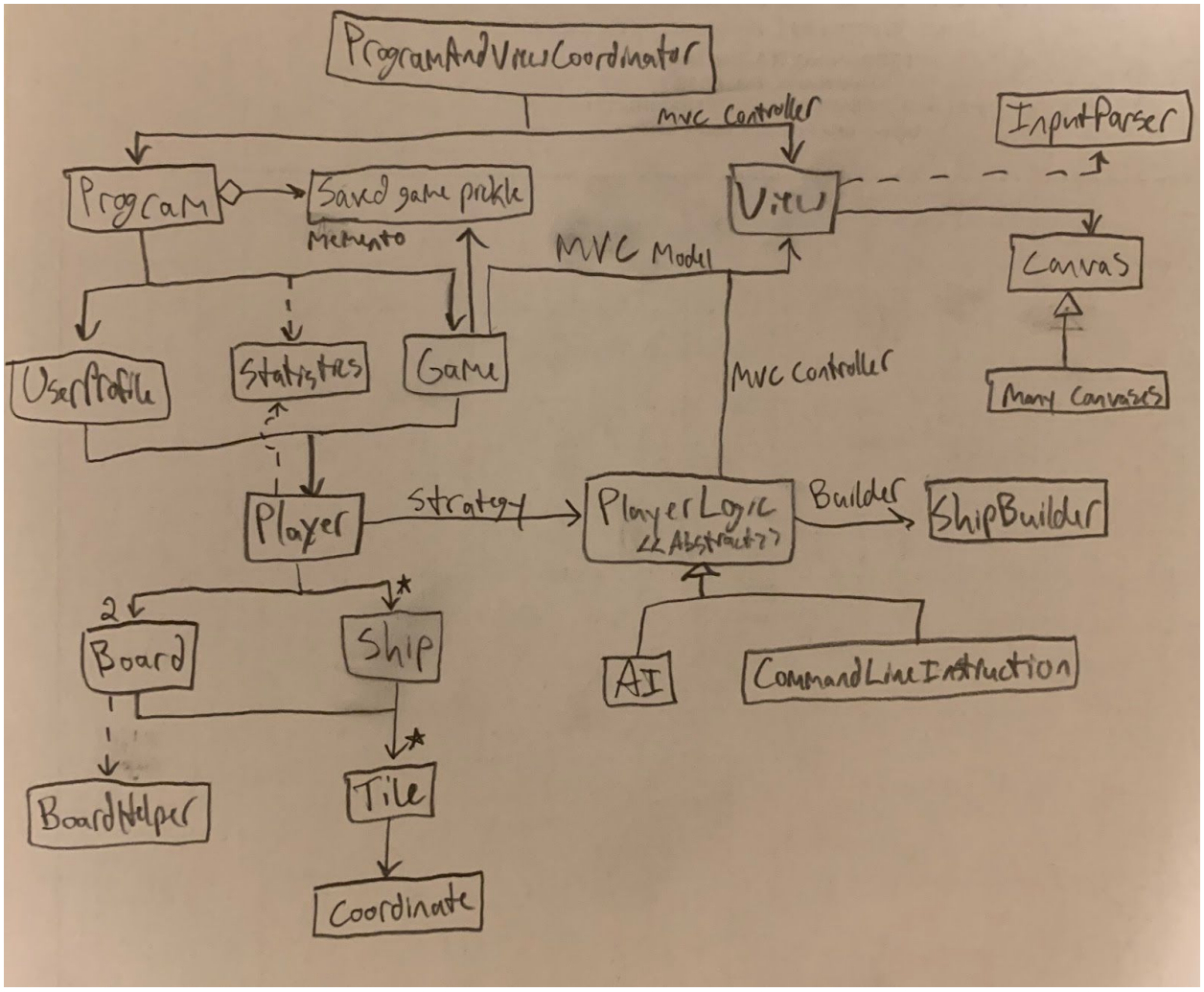
Original Class Diagram from Project 4



Class Diagram from Project 5



Class Diagram for Project 6 (Final Project State)



Note: We use more helper methods per class than originally anticipated. For clarity of the UML diagram, we have elided those methods and included only the main class names.

Key Changes From Original Plan

- ShipFactory was changed to the Builder pattern since ship instantiation requires multiple function calls and input from the user
- Ship was made simpler (only one class, only one kind of attack). Even if we had added special attacks, we would have kept it as an attribute in Ship rather than use separate classes.
- Addition of memento pattern for game saving and loading
- GameMode is extraneous since we use only one game mode
- We have only one kind of AI, so we did away with the AI factory
- Addition of ProgramAndViewCoordinator to simplify interaction between Program and View
- Separation of helper methods into separate, static classes (BoardHelper and InputParser)

- Statistics made static for easy reference

Third Party Code vs Original Code Statement

- https://www.cs.nmsu.edu/~rth/cs/cs187/f97/battleshipdesign.html
    - This source inspired the circular reference between Tile and Ship and the separation of Tile and Coordinate. This design makes it easy to localize updates to Board and pass messages (attacks) between players.
- https://gist.github.com/JungeAlexander/6ce0a5213f3af56d7369
    - This source provided a method for importing from a parent directory

Statement on the OOAD Process for Overall Project

1. The structure of this series of projects naturally lent itself to a Waterfall design process. This necessarily required us to make assumptions upfront about what we wanted to achieve and how the program would function at a low level. As you can see above, we made a good number of changes to our original design. We got rid of several elements when we determined that they weren't necessary to implement our desired program features. We also added elements (like the memento pattern) after we learned about it.
2. Parts of the project were written "test-during" and parts were written "test-after." The parts written test-during generally went smoothly and were generally error-free. The parts written test-after had lingering bugs that remained until proper tests were written for them or until we performed extensive manual testing.
3. Almost all of the coding was done individually. We had a few alignment meetings where we discussed high level work plans and then dispersed. It's a tough balance to strike. On the one hand, it's frequently more productive to code individually. On the other hand, complicated parts of the program can be more quickly understood with a face to face meeting. If we were to do it over, we might meet a couple more times with some face to face coding to understand those complicated parts more quickly.

Code Submission Link

https://github.com/jakeluoma/Battleship