Final Project

Released: March 28th

Preliminaries Due: Tuesday, April 16th at 5pm on Gradescope

Due: Thursday, April 25th at 5pm on Blackboard and Gradescope

No late days on this assignment! An automatic grace period until midnight for all deadlines will be given

You may work in groups of up to four

1 Introduction

This project will have you implementing the game Battleship in an application that can be played over the network.

2 Battleship

Battleship is a two player boardgame played where players set their boats up in a preliminary setup round and then alternate by choosing grid squares. If there is a boat in the grid square, it is a hit, and that is reported to the other player. Otherwise, it is a miss. Play continues until one player has hit all tiles of the opposing player. The rulebook for the game is available at the end of this document.

3 Your work

Work for this project will be submitted as three components. A planning document will be due on April 11th to gradescope. Code will be due to blackboard on April 25th and a final report will be due on April 25th to gradescope. All submission times are at 5pm. There are no late days for this project, but all deadlines have an automatic grace period until midnight the day it is due.

3.1 Preliminary Team Document

All projects will have a preliminary planning document due to gradescope by April 11th at 5pm. In this you should have the following. **Teams must be formed by this deadline and teams must submit together. Teams will not be allowed to be formed after this date**

a) A plan for how the work will be fairly divided among the team members with a hand-written signature from each member.

b) Wireframes for your client Battleship app.

c) State diagrams for both the client and server code.

3.2 Code

Your submission should contain two zip files containing maven projects: one for the server and one for the client. At a minimum, your code must support the following behavior

a) A Client application that allows you to connect to a server to play against another human opponent or against an AI. Your code may assume that there is a running server to connect to. Users should be able to play multiple games and the correct game state should be maintained.

b) Server code, which may optionally contain a GUI, will match clients against each other for play and keep track of the game state for each pairing. Server code will always be run before a client connects.

3.3 Report

For your final submission, you will create and submit a report detailing all work done above-and-beyond the requirements. It should include screenshots and a description of any features that are implemented outside of the minimum requirements. Additionally, you should report what each team member was able to complete and what work was done using outside resources, whether through other classmates, online resources or generative AI. The paper should be between 4-6 pages.

4 Academic Honesty

For this project **and this project only** all traditional academic honesty provisions are suspended. In particular, this means communication and sharing of code between peers is acceptable, and that ChatGPT and similar tools are fair game for use. For any behavior that does not conform to normal restrictions on academic honesty, there should be a discussion of what tools / resources were used and for what in your report.

5 Grading

This project is to judge your ability to create a complete project and see advanced implementation of features through to completion. The grade breakdown will be as follows:

- Preliminary Team Document (20%)
- Minimum Code requirements (50%)
- Report and above-and-beyond requirements (30%). *Completing the minimum required components of the assignment and the report, will earn a* 10/30 *for this portion.* It is up to you how to show your knowledge outside of the scope of the class and also up to how you represent these features in a clear and concise manner.

BATTLESHIP®

For 2 Players

OBJECT OF THE GAME

Be the first to sink all 5 of your opponent's ships.

CONTENTS

- 2 game units
- 10 plastic ships
- 4 runners of white pegs
- 2 runners of red pegs
- label sheet

THE FIRST TIME YOU PLAY

Apply the "Battleship" labels from the enclosed label sheet to the lids of the 2 plastic game units. See Figure 1.

Separate the 10 plastic ships from the runner. Each player's fleet contains 5 different ships (see Figure 2). Discard runner.

Twist red and white pegs off runners and place them in the peg storage compartments of each game unit (see Figure 3). Discard runners.

PREPARE FOR BATTLE

You and your opponent sit facing each other, with the lids of your game units raised so neither of you can see the other's ocean grid.

Secretly place your fleet of 5 ships on your ocean grid. To place each ship, fit its 2 anchoring pegs into 2 holes on your *ocean* grid. Your opponent does the same.

Rules for placing ships:

- Place each ship in any horizontal or vertical position, but not diagonally.
- Do not place a ship so that any part of it overlaps letters, numbers, the edge of the grid or another ship.
- Do not change the position of any ship once the game has begun.

See Figure 4 for an example of ship placement.

FIGURE 1

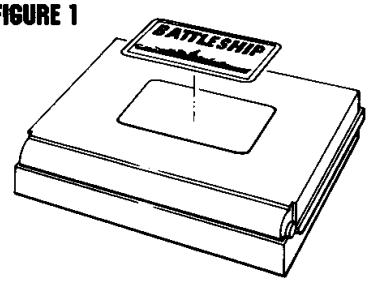


FIGURE 2


CARRIER—5 HOLES


BATTLESHIP—4 HOLES


CRUISER—3 HOLES


SUBMARINE—3 HOLES


DESTROYER—2 HOLES

Your fleet of 5 ships.

FIGURE 3

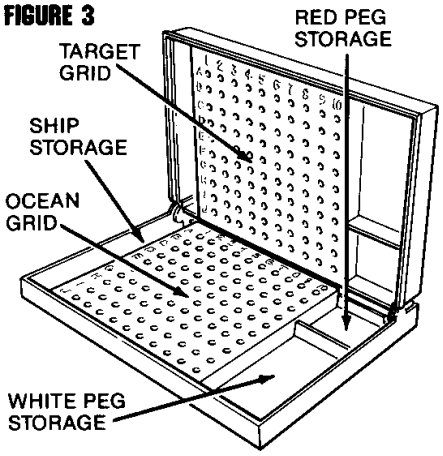
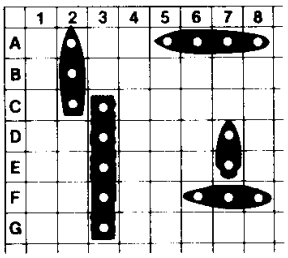


FIGURE 4



Here's an example of how to position your fleet correctly.