

File Organization 2025 Course Lab Project Assignment

You will create a one sided Battleship(Amiral Battı) game using only text based file management.

The Battleship tabletop game is a classic strategic naval combat game designed for two players. Here's a concise explanation:

Setup: Each player places their fleet on their own grid, with ships positioned horizontally or vertically without overlapping.

Game play: Players take turns guessing the positions of the opponent's ships by calling out coordinates. Hits and misses are marked on each player's board.

Winning Condition: The objective is to sink all of your opponent's ships before they can do the same to yours. A ship is considered sunk when all its parts have been hit.

Strategies: Effective strategies include focusing on areas with potential clusters of ship parts and employing a random approach to avoid predictable patterns.

Here is an exemplary tabletop Battleship game picture:



Above example is for one player, each player has a board which he can place his ships; which is horizontal in the example and a board for cannon shots aimed to the opponent's ships. So two players playing the game have one for each player. In our assignment you will have 2 boards for separate players: one for first player's ship board, the other board is the 2nd player's aiming board. Each board will be addressed like the above picture:

-The boards will be 10x10 matrix.

The columns are addressed by numbers from 1 to 10.

The rows will be addressed by letters from A to J

So for example, when placing the ships or aiming a bomb towards ships, user enters the address like F6 (first character is row F, the second character is column 6).

It's up to you how to format your text files but you need:

1. 2 text files named 1stships.txt and 2ndaim.txt (you need to save each board as the actual board placement)
2. You need to ask user to place their ships by asking first which ship to place;

They need to place (A for each) one 5 places long, (you need to ask 5 coordinates)

(B for each) one 4 places long, (you need to ask 4 coordinates)

(C for each) two 3 places long, (you need to ask 3 coordinates)

(D for each) two 2 places long, (you need to ask 2 coordinates)

(E for each) Three 1 place long ships. (you need to ask 1 coordinate)

Every time you need to ask rowcolumn address (ex F6) (you may ask all coordinates of **a ship** at once)

3. After the placements request from the second user to send cannons one coordinate only and update each file after each attack (We will control after each turn the files updated or not? (don't write to files in binary form in order to be checked directly after each turn))

Project Rules:

All projects must be uploaded to the classroom assignment page until 23:59 on Jun 07, 2025. Late submissions will not be accepted and the student's project grade will be 0.

Projects must be completed individually. If a project is found to have been copied from another source or if it is discovered that one student has shared their project with others, all students involved will receive a grade of 0 for the project. The grade will be based on the efforts of the student who actually completed the project.

Projects will undergo plagiarism checks. Those found not to be plagiarized will have their upload dates and times listed on the classroom page.

If a student is unable to present their project due to unforeseen circumstances, they must inform via the e-mail address:

demircid@iuc.edu.tr

At least one day before of the scheduled presentation date with an explicit request for a new slot. The new slot should not be later than the originally assigned date. Students who do not present their projects on time will receive a grade of 0.

Projects can be developed in any desired programming language.

Projects must be uploaded to the system as rar or zip files with the student's number_name and surname. (Example: 1306250001_Ali Veli.rar) Those who do not adhere to this rule will receive a grade of 0 directly. **The compressed file should not contain folders or any other necessary files to compile etc, only the specified files.**

A) A file with the exact same name as the compressed file for the source code (example: 1306230001_Ali Veli.c)

B) An executable .exe file if possible with the language used (The name of this file must match the name of the source code file) C) Two text files having an exemplary game updated or at least the starting status of the game Text files should be located in the same folder as the exe file for running compatibility purposes."

