

Programming Fundamentals (CS1002)| FALL2022

Due Date: 9th December, 2022 7:00am on GCR

Total Points: 100

INSTRUCTIONS

- 1. Plagiarism in course project will result in F grade in the course**
2. This is not a group project and each person will be working on the project individually.
3. Make sure you submit your project before the submission time. Late submissions won't be accepted even if they are late by just one minute.
4. You can earn bonus marks by implementing extra features in the project.
5. Use good programming practices (well commented and indented code; meaningful variable names, readable code etc.).
6. Each file that you submit must contain your name, student-id, and assignment # on top of the file in comments.
7. Combine all your work in one folder and compress it into a zip file. The folder must contain .cpp files (no binaries, no exe files etc.).
8. Submit the solutions via google classroom. Submissions via email will not be accepted.
9. Use proper naming convention to name the file containing source code.
E.g. i22xxx_project.cpp , replace i22xxx with your roll number.
- 10. Follow the given instructions to the letter, failing to do so will result in a zero.**

Battleship Game

This project is an extension to the battleship game that you have already developed in your 5th assignment.

Battleship is a two-player game, in which the player defeats his/her enemy by sinking his/her fleet. You place your ships wisely and destroy your enemies' ships. In this game strategy as well as luck both play part.

Sea Battle 2 is similar version (mobile version) of this game. You can play this game to understand the rules of the game.

You are required to make two-player Battleship game, where one player is computer. You have to place your ships over the grid and wait for your opponent(computer) to do the same. Once the game has started you have to indicate the coordinates of your shot by clicking on the location (cell) on the grid. If your shot is successful, a symbol appears telling you to keep trying your luck around that area to sink that specific ship. Game will continue until all battleships of any of the player are destroyed.

In this project, you have to make the Battleship game with the following features:

Game Start Menu

- a) When the game starts, a user should be given the choice in the menu to
- **View the leaderboard.** (displaying top 10 scores along with player names). If the game is being played for the first time the leaderboard will be empty. Otherwise, the high-scores and player names should be read from the file (named highscores.txt) on the hard-disk and stored in the arrays of size 10. If the score of the current game makes to the list of high-scores, the lowest score is removed from the array and the new contents of the array overwrite the file.
 - **Start New game.** Once a player has selected this option, he/she will enter following
 - Player's name
 - Color of the battleships
 - **View Balance.** For an existing user, it should provide an option to view available balance.
 - **Options.** Player can select mines, submarine, bombers. You can display these options on the game board as well.

The Board

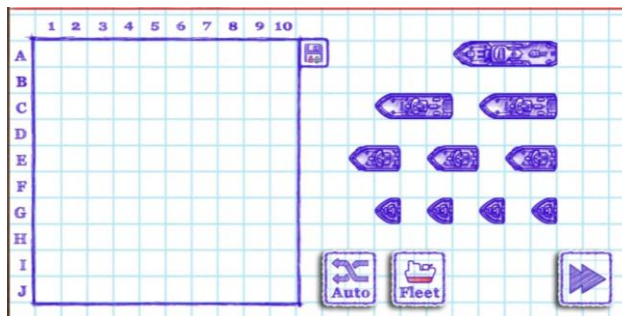
- b) Draw a $n \times m$ board on the screen. Board has two areas, a game grid and options area. You can set the dimensions of the game board as per required. For example, you can set them to 14×24 . Memory for the board will be allocated dynamically. However, dimensions are static, i.e., not taken by the user.
- c) Draw a 10×10 game grid (battle sea) on this board.
- d) Draw battleships of various lengths on the right side of the board. Details of battleships are as follows

One battleship of length 4
Two battleships of length 3
Three battleships of length 2
Four battleships of length 1

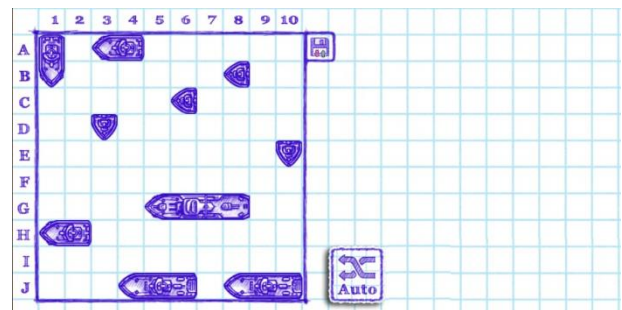
You will show the length of a battleship with number of cells it occupies on the grid.

- e) Player will place battleships on the 10x10 grid. For this player will select a battleship and then the starting cell on the grid. Battleships will be placed facing towards left. For example, if you want to place battleship of length 3 in the cell E7. You will click on E7, and it will occupy E5, E6 and E7. You are required to do input validation and display appropriate message.
- f) Orientation of the battleship can be changed by clicking and moving mouse.
- g) Rules for placement of battleships are as follows
 - No two battleships can be placed in the same cells. There should not be any overlapping of the battleships
 - There must be a distance of at least one cell between each battleship.
 - Orientation of battleships can be different. They can head left, right, up or down of the grid.
- h) You will also provide an option of **AUTO** on this screen. By selecting this option, automatic(random) placement of battleships will take place.
- i) Additionally, you can display scores, balance, armory and timer.

A sample board is attached below.



Before placement of battleships



After placement of battleships

Game Play

- a) Once the game has started, both of the grids(player and computer) will be displayed. (both 10x10 grids). However, position of battleships of computer must not be shown. Player and computer will play turn by turn. On their turn, each will shoot at some empty cell of the opponent's grid.
- b) Player will take a turn and click on the cell to shoot. There are two possibilities
 - a. If there is enemy's battleship, display 'x' in the cell to indicate that battleship has hit and take another turn. (fig 2a)
 - b. Otherwise display 'b' and turn will change.

- c. If battleship is completely destroyed, i.e., all cells covered by a battleship have been shot, the area around the battleship (cells adjacent to battleship) will be revealed, i.e., marked with b. (fig2c)
- c) Player who will destroy the opponents' battleships first would be a winner.(Fig 3a)
- d) If a player fails to take his/her turn within 10 seconds, turn will change automatically.
- e) At the start of the game, each player will have a balance of 500 only for the first time.

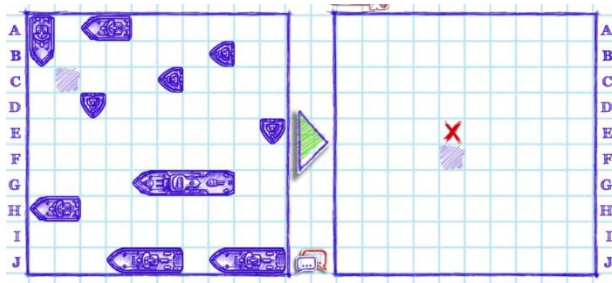


Fig 2a

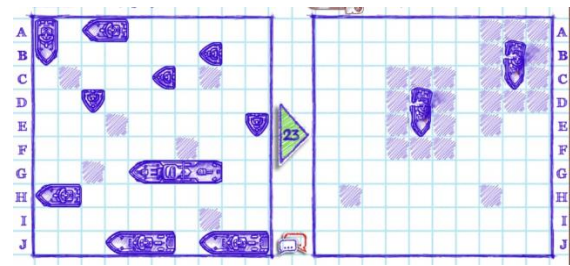


Fig 2c

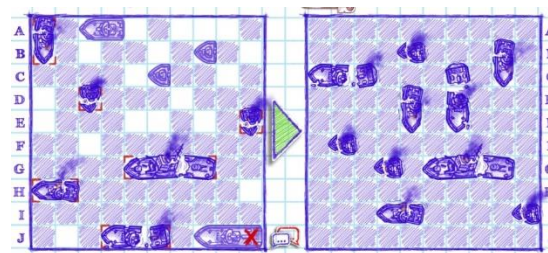


Fig 3a

Scoring

- f) For each successful shot, you will gain 60 points. If the distance between two successful shots is more than 2 turns, then there would be negative points of (-20) for each unsuccessful shot.
- g) At the end, these points will be added to the available balance.

Enhanced functionality (30% of total points)

- a) Translation and/or Rotation can be applied to the battleships after placing them on the grid. Battleships can only be rotated in anti-clockwise direction. Player is not allowed to translate/rotate battleship once the game starts.
- b) You can also purchase additional items (armory) and use them. You can place mines on your board whereas you can use bomber and submarine to shoot at the enemy's grid. If you have purchased any of the items below, then display them (Bomber and submarine) on the side of the board whereas mines on the board, so that they can be used. You should modify the available balance displayed on the board as well as in the file (saved data) for the next time. Details of the items, their count(that you can use in one game) and cost of each is given below

Item	Function	Cost/item	Count
Mines	When hit by the enemy, cell on his grid will be shot too	100	5
Bomber	Will destroy the whole battleship, if any of its cell is in the way of bomber. At a time, it will destroy only one battleship, i.e., one with low column number.	200	2
Submarine	You can hit range of values (any three consecutive cells). It will destroy all ship that are in first and last column	250	1

- c) For the placement of mines, player will specify the location (cell on the grid) by clicking on it.
- d) You can use bombers to shoot at the enemy's grid. You will select the bomber and then click on the location on the grid to shoot. It will destroy the entire row. For example, if you click on D5, entire row D will be destroyed. If there is a battleship of length three in the cell, "B3" with orientation down. Only one cell, i.e., D3 will be destroyed. In this case, for each successful hit, you will get 40 points against each occupied cell. According to given example, you will get 40 points only.
- e) You can also use submarine to destroy your enemy's grid. You will select the submarine and then click on the location on the grid to shoot. It will destroy the entire row and column as well. If there is a battleship of length three in the cell, "B3". Only one cell, i.e., B5 will be destroyed. In this case, for each successful hit, you will get 40 points against each occupied cell. According to given example, you will get 40 points only.

Helper Functions:

We have provided you the code (helper functions) in C++ to draw shapes (circle, square and triangles) of different colors and sizes. You are allowed to change it and add functionality according to the project statement.

To execute the starter code, you need to do the following:

- a) Extract the attached zip file.
- b) Open the terminal and navigate to the path of extracted directory
- c) Install the required libraries by executing the command below:
`bash install-libraries.sh`
- d) Compile the project by writing the command
`make`
- e) Run the main file
`./game`

Important Note:

You must use the concepts of functions, pointers, dynamic memory allocation and file handling that have been taught to you in the course.

Happy CODING 😊