Battleships: Project Guidance (rev1.0)

November 18, 2022

1 Introduction

We have been answering some questions about the ense352 fall 2022 project and thought we should share some information with you all to provide some clarification and guidance.

NOTE: the following is is not a set of prescriptions on how everything *must* be done, rather it's an example and a starting point.

For example, you are strongly encouraged to use a paper-base *targeting grid* to track the shots you make on your oppponent (and the results of those shots), but you could use any equivalent technique, like including a LCD or console display, morse code, or FAX. Ok we're kidding about the last two.

Where you *do* have to constrain your design is this: your system must be compatible with other players systems:

- The shot coordinates must be called out, by the shooter, as a *pair of numbers* in (row, column) order.
- the software on the *victim's* end (the party being shot at), after receiving the shot coordinates, must automatically recognize all four types of shot event (described below).
- you must *as a mimimum* support the standard 10x10 grid and 5 vessels of size 5, 4, 3, 3, and 2.

2 Game Setup

When you play Battleships against another player, you will use your **game unit** (the nucleo64 board, extra hardware, and software you created) as your battleship device, and your opponent will use their **game unit** as their device.

When you start a game you'll "place" your ships on the game board by

1. editing your source code and adding a number of **Vessels** (generally five of them), and specifying the type of Vessel, and the grid cells occupied.

2. choosing the size of the grid representing the "ocean" (standard Battleships has a 10x10 grid)

3. then you'll recompile your software, flash it, and run it. This will be necessary before **every** game. We don't expect you to be able to set up a new configuration in a running system.

    - this means your opponent will have to do the same before starting to play against you. They must use the same size of grid and the same number of ships, to make it a fair contest.

3 Game Play

1. Each player will have a **game unit**, a pencil, and a paper grid of the correct size on which they track their shots (the **targeting grid**).

2. To start the game each player can press the blue **User** button on their f103rb board.

3. You and your opponent can decide on who goes first, after ensuring you have the same number and type of vessels, and the same grid size. Let's call the starting player **player1** and the other **player2**, and the corresponding units **game unit1** and **game unit2**.

The game now proceeds as a series of rounds. Each round consists of two phases. In phase 1 (**P1**) **player1** has the role of **attacker** and **player2** has the role of **victim**. In phase 2 (**P2**) they exchange roles. The next steps detail the two phases of a single round.

P1-1 **Attacker** calls out a grid cell location in **row, column** format. For example, targeting row 3 and column 5, the **attacker** calls out **3,5**. Notice first the *row* is stated, then the *column*, just like in Linear Algebra.

- The **attacker** must record this shot, by annotating their **targeting grid**.

P1-2 The **victim** enters this location into their **game unit** as a pair of numbers, using a combination of DIP switches and button presses.

- The **victim's game unit** records the result of the shot, and notifies the **victim** of the result. There are 4 possible results, or "events": **miss**, **hit**, **sunk**, and **all sunk**.
- The **victim** verbally informs **attacker** of the event type. The **attacker** must record the event details: we recommend using the **targeting grid** for this purpose. The **victim** has nothing else to do: their **game unit** has already recorded the event.
  - If the event was **all sunk**, *most* of the time the game is over and **player1** has won. The one exception: if **player2** is one shot away from sinking **player1's** last vessel, the round must complete. If **player2** also gets the **all sunk** event the game is a draw, otherwise **player1** has won.

P1-3 This ends **phase1** and begins **phase2**. **Player1** and **player2** switch roles: **player1** becomes the **victim** and **player2** becomes the **attacker**.

P2-1, P2-2 these are the same as in **phase1**, with of course the roles interchanged. If the **all sunk** event occurs, and

- the **all sunk** event also occurred in **phase1** then the game is a draw, otherwise
- the game is over and **player2** has won.

P2-3 play continues to the next round, going back to step **P1-1**.

4 Revision History

Date	Details
2022-11-18 Fri	Initial revision