NewFeatureRequirements

Created by	 Paul Weaver
Created time	@March 3, 2024 10:54 PM
Tags	BattleshipLite Everything

Basic Requirements

We have a full working Battleship clone application that currently supports a 5×5 (A1-E5) grid. It is now a request to build a 7×7 (A1-G7) grid for more possible game scenarios and lengths. In addition, they would like this to be a customizable option for the user, meaning they should have the option to use the same standard 5×5 grid or the new 7×7 grid.

General Flow

We use the same general flow as mentioned in the **ActualRequirements** document, but after the user opens up the console, they are asked a message to select the grid size before the game starts.

- Two users open up the console
- **Ask for grid-size**
- Ask user 1 where to place their ships
- Ask user 2 where to place their ships
- Ask user 1 for a shot
  - Determine hit or miss
  - Determine if the game is over
- Ask user 2 for a shot
  - Determine hit or miss

    - Determine if the game is over
- Repeat until someone wins
- Identify who winner is
- Exit the application

Additional Questions/Requirements

- Is this feature enabled by the user?
    - **Yes**
- Can they customize the grid size to however large they want?
    - **Maybe**. That is not something we want to explore as of now. Perhaps in the future.
- Currently, we tell the user how many shots it took for them to win. It will say “User took 5 shots.” Will we change that text to 7 shots if it’s a 7×7 grid?
    - W**e will still determine a winner with 5 ships and shots and not 7**. That is a definite possibility in the future.

Full Requirements

1. 2-player game
2. 25 spot grid (A1 - E5)
    a. **OR: 49 spot grid (A1-G7)**
    b. **User chooses which grid size they want**
3. Each player gets 5 ships
4. Each ship takes up one spot
5. Players take turns firings

6. First person to sink all 5 wins
7. One console for everyone
8. No completing the round after 5 sunk ships
9. Show a visual grid with hits and misses
10. Do not allow the user to shoot the same spot twice

UI Design

Welcome message

Ask the user to determine if they want a 5×5 or 7×7 grid

    it doesn’t matter if user 1 or user 2 or both decides

Ask user 1 for their name

Ask user 1 for their 5 ship placements

    Ask for placement

    Determine if it is a valid spot

    Store it

    Clear

Ask user 2 for their name

Ask user 2 for their 5 ship placements

    Ask for placement

    Determine if it is a valid spot

    Store it

    Clear

Display grid of where user 1 has fired

Ask user 1: Where would you like to fire on?

    Verify a valid spot

Check results

Store shot

Clear

Display the score (user 1: 2 ships left, user 2: 4 ships left)

Repeat with user 2

Loop until someone wins

Print out winner’s name and number of shots taken

Wait for user to say they are done

Exit

UI Design cont.

If user at the beginning uses a 5×5 grid:

A1 A2 A3 O A5

B1 B2 B3 O B5

C1 C2 C3 X C5

D1 D2 D3 O D5

E1 E2 E3 O E5

If user at the beginning uses a 7×7 grid:

A1 A2 A3 O A5 A6 A7

B1 B2 B3 O B5 B6 B7

C1 C2 C3 X C5 C6 C7

D1 D2 D3 O D5 D6 X

E1 E2 E3 O E5 E6 E7

F1 F2 F3 O F5 F6 F7

G1 G2 G3 O G5 G6 G7

Logic Design

Clear Display

- Let's say you've never done a clear display before. This can apply to anything in your logic design, but in this example, you've never done a clear display before. If so, one of the things he recommends is to create a dummy application and print something to the screen (in this case). And below you Hello world print statement, you might try different commands to clear that hello statement off the screen. Once you find something that works, you'll put it in place and run the application to make sure it works.

Enum?

Method: Ask for Grid size

Method: Asking for name

Method: Get ship placement

- Both methods keeps code DRY as it can be called by user 1 and user 2 in this battleship clone

Method: Determine if valid spot for ship

Storing ship information: List per user?

Storing shot information: List per user?

Method: create the grid for each user

- **There should be an if statement to check if the game is using a 7×7 grid or normal 5×5 grid.**
- **Based on that if check, that grid will be used for both user 1 and user 2.**

Method: print out grid

- **I don't anticipate this method to change, but I could be wrong. It should simply print out whatever grid gets initialized. Either 5×5 or 7×7.**

Method: fire on opponent

Method: determine if a shot can be taken & outcome

- Ex. Let’s say you want to shoot at A4. If it was already determined to be a miss, no you can’t shoot there again because there’s no point. But A3 on the other hand hasn’t been shot at. Can be hit or miss

Method: display score

Method: print winner and shots taken

Data Design

Nothing should change in the data design since we’re not needing to store new data about the grid or player.

PlayerInfo

User’s name - string

User’s 5 ship locations - List<GridSpot>

User’s shot grid - List<GridSpot>

GridSpot

SpotLetter - string

SpotNumber - int

Status - string (possible enum)