2.1

- Ships which can be placed on a grid, you should be able to move and rotate (90 degrees) your ships within the grid before the game starts.
- Player grid pattern with interactable tiles in the grid
    - Ships to be placed on the grid with a span of multiple tiles, check whether all the selected tiles for a ship are available to place the ship and if so place the ship
    - Check if tile is available to shoot at
    - If tile has been shot show

2.2

- Game state
- Player board
- Opponent board
- Ships
- Ship placement
- Cells that are hit/missed
- Who won the game
- Which ships are already destroyed
- Timer

Design pattern: MVC (model view controller)

3.3

- Client to server: game is complete, game won by …
- Server to client: abort game (one player left)
- Sever to client: place ships (may not be implemented, might use random ship placement instead)
- Player A to server or player B to server: I want to place my shot on this cell
- Server to player A or player B: shot hit
- Server to player A or player B: shot miss
- Server to player A or player B: player … destroyed one of your ships
- Server to player A or player B: you destroyed one of player …’s ships
- Server to both players: game over with result won/loss