W & O Steps

Monday, May 22, 2023 3:39 PM

Basic Requirements

- 2 player console game
- Based on Battleship from Mattel
- 25-spot grid (A1-E5)
- Each player places five pegs, representing their five ships
- Players take turn firing on opposing ships
- First person to sink all opposing ships wins

General Flow

Two users open up console
Ask player 1 for ship positions
Ask player 2 for ship positions

Ask player 1 to shoot
    Hit or miss?
    Are all ships destroyed?

Ask player 2 to shoot
    Determine hit or miss
    Determine if the game is over

Repeat until a player wins
Identify who wins

Exit applications

Additional Questions/Requirements

1. Do we use the same console or two different consoles working together? **Same Console**
2. Does the other player get martyrdom? (One last chance when getting hit) **No**
3. Do we want to get and set statistics (Hit/miss ratio)? **Only how many shots it took to win**
4. One ship goes to one spot.
5. Do we allow players to shoot the same spot twice? **No**
6. Do we show a grid visually? **Yes**
7. Do we store game data? **No**
8. Is the number of players going to change down the line? **Maybe**
9. Will we add an AI (Computer) Player? **Maybe**

Full Requirements

1. 2 Player Game
2. 25 Spot Grid (A1-E5)
3. Each player places 5 ships
4. Each ship takes up one grid spot
5. Players take turns firing
6. First person to sink all 5 wins
7. One console for both players
8. No completing round after 5 ships have been sunk
9. Show a visual of grid with all hits and misses
10. Do not allow the user to shoot the same spot twice

Planning is not always perfect. Try to identify the big picture
and ask good questions

U, L, & D Steps

Monday, May 22, 2023 3:47 PM

User Interface (UI) Design

- Welcome message

- Ask player 1 for their username
- Ask player 1 for the 5 ship placements
  - Ask for placement
  - Determine if this is a valid spot on the grid
  - Store data
  - Clear screen

- Ask user 2 for their username
- Ask player 2 for the 5 ship placements
  - Ask for placement
  - Determine if this is a valid spot on the grid
  - Store data
  - Clear screen

- Display grid of player 1 shots (nothing if first time)

- Ask player 1 - Which spot on the grid they would like to fire?
  - Verify valid spot
  - Check result
  - Store shot
  - Clear screen

- Display scores (Player 1 - 2 Ships left, Player 2 - 4 Ships left)

- Repeat this with Player 2

- Loop until a player destroys the other player's ship (wins the game)

- Print out winning player name and number of shots taken

- Wait for user to prompt exit
- Exit

UI Design (Grid)

A1	A2	A3	A4(O)	A5
B1	B2	B3	B4(O)	B5
C1	D2	C3	C4(X)	C5
D1	D2	D3	D4(O)	D5
E1	E2	E3	E4(O)	E5

Logic Design

Clear display

Method: Asking for username and ship placements

Method: Get ship placement

Method: Determine if valid spot for ship

Storing ship information: List per player?

Storing shot information: List per player?

Method: Create the grid for each user
Method: Print out grid

Method: Fire on opponent

Method: Determine if shot can be taken (hasn't shot at this spot already)
& outcome

Method: Display scores

Method: Print winning player and shots taken

Data Design

Player's Name
Player's 5 ship location
Player's grid shots

GridSpot
SpotLetter
SpotNumber
Status - possible Enum (string initially)

Planning out your data and UI ahead of time helps making the application easier.