Day60 Battleship

Write a Battleship game that uses a 5 row by 10 column 2 dimensional String array to hold a game grid. Make each location equal to a tilda "~" to start, then place a capital "S" in a random location to represent the ship. Then ask the user repeatedly to guess the row and column of the ship. If they get it right, say "You sunk my battleship!" and end. If they get it wrong tell them that they missed and put an X in that location. Print the board before each turn. If the user enters values out of range, tell them instead of crashing and let them re-enter values. Write and use a method to print your String array.

Don't make any part of the program hard-coded to the size of your grid except for the line where you declare the String array. In other words, use grid.length and grid[0].length everywhere you need to refer to the size of the array, not 5 and 10. In a real Battleship game you wouldn't print an S where the ship was, but we don't want to spend hours testing this, so we're printing the S.

Please test your program by firing some intentional misses. Does an X appear for each one? What happens if you enter a value that is out of range for either the row or column? Make sure your program does not crash. Then, actually hit the ship to make sure everything is working before you think you are done.

Name your project Day60 Battleship and turn in a share link on the Google Classroom when you are done.

Checklist:

- Use the integers 5 and 10 when you create the grid array and then use grid.length and grid[0].length anywhere else that you work with the grid.
- An S is placed randomly using Math.random() calls at the beginning of a run.
- You write a method which prints the board looking like my examples below.
- A turn looks like this: the board is printed, then the user is asked to choose a row and column. If the ship is at that location, you print "You sunk my battleship!" and the game is over. If it is not, you put an "X" in that location and tell the user they missed.
- Your code checks for entries out of range (below zero or over the # of rows or columns) and reports a clear error instead of crashing. See sample output below.

Sample output below.

```
Welcome to Battleship!

~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ S ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Enter a row:
 3
Enter a column:
 3
Miss!

~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ S ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ X ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Enter a row:
 20
Enter a column:
 -4
One or more entries is out of range. Please try
again.
```

```
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ S ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ X ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Enter a row:
 2
Enter a column:
 2
Miss!

~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ S X ~ ~ ~ ~ ~ ~ ~
~ ~ ~ X ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Enter a row:
 2
Enter a column:
 1
You sunk my battleship!

~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ @ X ~ ~ ~ ~ ~ ~ ~
~ ~ ~ X ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```

Optional Extra credit: add a "laser" feature which Xs out an entire row, and/or add a nuclear option which Xs out a 3 by 3 square all at once. Use -2, -2 as the row and column to activate the laser option and -1, -1 to activate the nuclear option. See sample run below.

If you do the extra credit, turn in your regular project first, then fork your repl and do the extra credit there, send me the link for the extra credit in an email. (I won't be checking the main submission for extra credit features.) Thanks.

Sample run with extra features:

```
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ S
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
Enter a row:
-2
Enter a column:
-2
You have activated the laser. Enter the row:
0
X X X X X X X X X X
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ S
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
Seriously, even with a laser you missed?
X X X X X X X X X X
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ S
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
Enter a row:
-1
Enter a column:
-1
You have chosen the nuclear option.
Enter a row:
3
Enter a column:
8
X X X X X X X X X X
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ X X X
~ ~ ~ ~ ~ ~ ~ X X X
~ ~ ~ ~ ~ ~ ~ X X X
BOOOMMM! Got it!
```