Dice Roller

Problem Description:

The client; DevsInc, has requested production of software that allows users to get random numbers between two values, simulating the roll of a die with a set number of sides. The interface must allow users to select how many sides their die has, and display the result of the die roll clearly.

User Story:

A tabletop game player; Sally, decides to use this solution to replace the use of a physical die in her game. She opens the UI and selects the number of sides she needs for the die. A button clearly labelled ‘Roll’ is present on the screen, she clicks it for her result. A random number within the limits of her selected sides is then clearly displayed and labelled as the result of the die roll.

Unit Test:

Testing the end solution can be achieved by isolating the function that generates the random numbers, running it many times and viewing the results to ensure that the numbers produced are actually random.

Solution:

Landing page:

Result from 20 sided die:

The solution UI is clearly titled ‘Dice Roller’ to represent its function. A six sided die is the default but the user is prompted to enter how many sides they need their die to have - up to

20 sides. A button labelled ‘Roll!’ is present to display the result. Once this button is clicked, the random number is displayed in a large font in the middle of the screen.

Test Result:

In order to verify the random number generator, the function is isolated and run many times in a ‘for’ loop, and the results of each generation are saved.

```
function diceRoll(sides) {
    min = Math.ceil(1);
    max = Math.floor(sides);
    return Math.floor(Math.random() * (sides) + 1);
}

var results = []

for (let i = 0; i < 10; i++) {
    results.push(diceRoll(20))
}

console.log(results)
```

10 runs of 10 generations between 1-20 are displayed:

1. 6, 10, 5, 10, 13, 12, 2, 17, 16, 2
2. 20, 9, 9, 9, 11,14, 16, 15, 14, 4
3. 19, 19, 1, 16, 5, 12, 17, 2, 6, 9
4. 11, 10, 4, 11, 20, 14, 4, 14, 4, 8
5. 15, 10, 12, 7, 15, 9, 6, 10, 7, 4
6. 20, 7, 7, 16, 6, 2, 7, 3, 9, 15
7. 1, 18, 11, 5, 17, 6, 8, 6, 1, 2
8. 1, 3, 19, 6, 15, 17, 15, 10, 10, 14
9. 9, 18, 20, 13, 10, 10, 16, 13, 5, 5
10. 3, 2, 17, 15, 4, 5, 14, 18, 14, 6

With all results displayed, it can be verified that the numbers being generated are random and suitable for the solution.