JavaScript Observation Test Exercise v1

Background:

The nature tour company, Animal Spotters, is hiring tour guides. They have found that their best guides are easily able to spot animals from a distance. They want applicants to take an observational test that will eventually be used to pre-rank applicants prior to any interviews. Applicants will see a screen filled with numbers. Their job is to click the numbers in order from 1 to the max. Animal Spotters has asked you to build out a proof of concept for this screening tool.

Requirements:

1. App should use JavaScript, HTML, and CSS
2. The page should display a grid of buttons containing numbers from 1-25
3. The initial class name of any button containing numbers must be “tile” with a hex code of rgb(27, 62, 119) and this color may NOT be used anywhere else in the exercise.
4. Numbers should have a randomized order
5. When a user clicks a number, if it is the next number in order, then its background turns a different color for the remainder of the test. If it is the wrong number, nothing happens
6. Once all numbers have been clicked in order, display a win message with the text “You Win!”.
7. Include a button to start a new game at any time with the text content “Start!”.
8. Hitting the “Start!” button defaults to starting a game containing 25 numbers and clicking the start button on subsequent clicks defaults to starting a game containing 25 numbers and will shuffle the numbers and restart the game. No other buttons to start a game are allowed.

Technical Requirements:

- All event listeners should be named functions
- No global variables
- All styling in CSS file
- All events and functionality in JavaScript file
- Multiple functions

- JSDoc Comments on all functions
- No use of console.log, alert, prompt, or confirm

Stretch Requirements (These are optional requirements for additional practice):

1. User can select the max number to count to
2. The win message now includes the total time taken (minutes and seconds) to finish the test.
3. The timer is now always displayed, and should update every second.
4. When a user clicks a wrong number, then 10 seconds are added to the total time
5. When a user clicks a wrong number, shuffle all remaining numbers
6. The Observation test now has multiple versions. Implement however many you want. Allow the user to select which option to attempt.

   6.1. **Doubles:** Numbers will be displayed in doubling order up to 25 or user-specified max, and must be clicked in that order. (1,2,4,8…)

   6.2. **Fibonacci:** Numbers will now be in the Fibonacci sequence up to 25 or user-specified max, starting with 1,2, and must be clicked in that order. (1,2,3,5,8,13…)

   6.3. **Factored**: Numbers must be clicked in order from fewest prime factors to most prime factors. If two numbers have the same number of prime factors, the lowest number should be the next one clicked.

   Example: Given a max number of 5, this is the order from fewest to most prime factors: (0,1,2,3,5,4). 0 and 1 both have zero prime factors. 2,3, and 5 are prime, so they have one prime factor. 4 has 2 prime factors (2,2), so it should be the last number clicked.
7. Numbers are displayed on top of a customized background. A user can upload an image file to be used as the background.