Assignment – React  
COS318 – Web Programming

Water. Earth. Fire. Air.

Long ago, the four nations lived together in harmony. Then, everything changed when the Fire Nation attacked. Only the Avatar, master of all four elements, could stop them, but when the world needed him most, he vanished. A hundred years passed and my brother and I discovered the new Avatar, an airbender named Aang, and although his airbending skills are great, he has a lot to learn before he's ready to save anyone. But I believe Aang can save the world.

In this assignment, you’ll be using React to create components representing the four nations that will create and destroy rocks that represent the battle of the lands. Let’s hope that the fire nation does not triumph.

1. **(10 Points) App component**
   1. The root App component created by create-react-app will control the state of your application. (the rocks themselves, as well as any other state you need)
   2. Create three div containers, one for the rocks, one for the four element components, and one to show the current number of rocks.
      1. The rocks and element containers should be styled with flex so the rocks and element buttons are centered on the screen.
      2. The rocks should not overflow the horizontal direction of the screen, i.e. they should wrap if there are more than can be rendered in a single row.
   3. The maximum number of rocks that should be rendered is 1,000.
2. **(30 Points) Rock component**
   1. Each rock component must render an image or SVG at a random size, small, medium, or large.
   2. Rock components will not change after they are created, so they should be exported using memo.
3. **(40 Points) Earth component**
   1. The Earth component will create rocks. It must render three buttons.
      1. The three buttons should create 1, 2, or 3 rocks respectively.
      2. If the number of rocks is already at the maximum, the Earth buttons should be disabled.
4. **(10 Points) Water component**
   1. The Water component will create rocks. It must render one button.
      1. The button will double the current number of rocks, but stopping at the maximum number of rocks.
      2. If there are zero rocks, or the number of rocks is already at the maximum, the Water button should be disabled.
5. **(10 Points) Air component**
   1. The Air component will remove rocks. It must render one button.
      1. The button will half the current number of rocks.
      2. If there are zero rocks, the Air button should be disabled.
6. **(30 Points) Fire component**
   1. The Fire component will destroy rocks. It must render one button.
      1. The button will destroy all of the rocks.
      2. If there are zero rocks, the Fire button should be disabled.
7. **(20 Points)** Code style, formatting, completeness, and quality.

Stretch Levels

If you already have a lot of experience with React or are an element bender, try to complete these stretch levels for a reputation bonus. If you try for the stretch levels, make sure to type it in the comments on Moodle so I don’t miss it.

**Appa Level**

Add some CSS to your page to make it look nicer. Background colors, font colors, or anything that looks good.

**Katara Level**

Instead of having the rocks only render at one of three sizes, make them render in at least 100 different sizes. (Hint: you won’t be using CSS classes for this.)

**Zuko Level**

Add an onClick event to the Rock component. When a rock is clicked, it will change it size, with a smooth transition to the new size.

The Rules

1. No inline styles or inline javascript.
2. Error messages must be “in-page” i.e. no pop-ups or alerts.
3. Any resources not created by you (images, javascript libraries, etc.) must be referenced using a CDN or URL, not directly included in your assignment submission.
4. All requests that submit a body to your server must have their entities validated with appropriate annotations, such as MinLength, Range, or Required.
5. The root path of your server must display the main page of your application.
6. Service/data/model classes must not have any http, request, or response references.
7. Controller entity classes must not be used directly to store data on the server; translate them into a model (data storage) class before saving the data. Conversely, controllers must not send any model classes to the user; translate them into controller entity classes before sending the response.
8. All service class instances must be obtained using dependency injection.
9. You may not use any synchronous methods in your C# code wherever there is an async option.