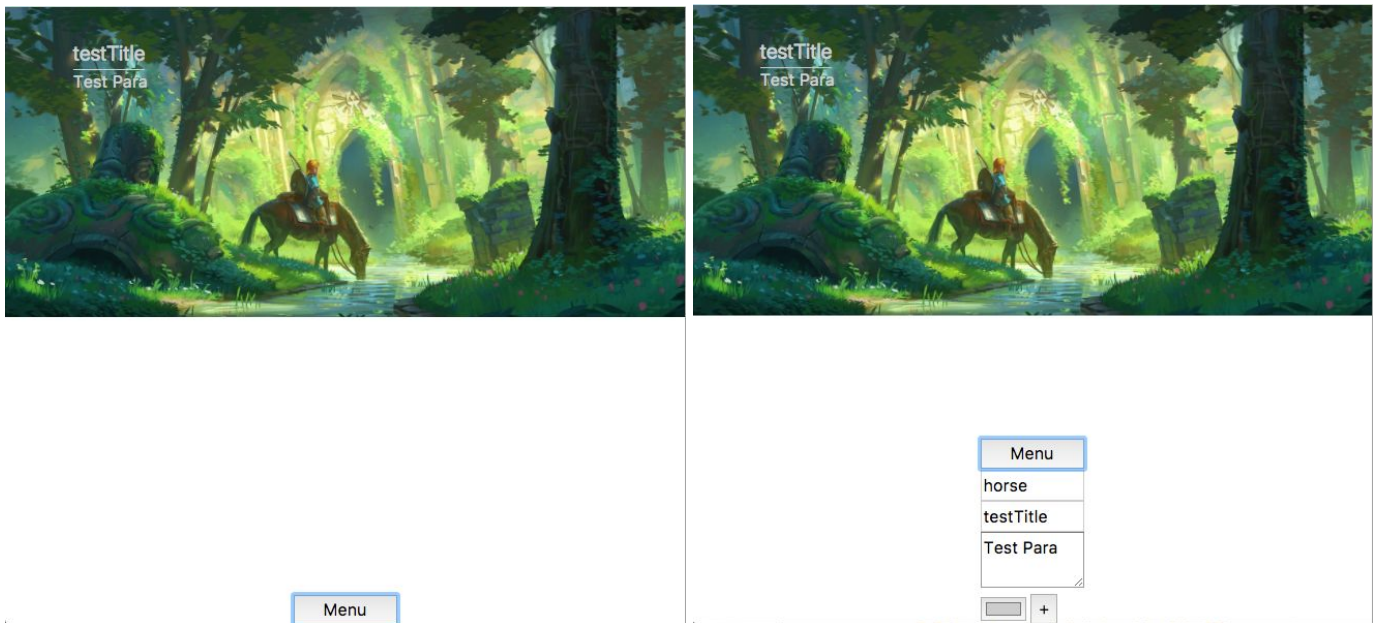A SIMPLE(?) STORYBOARD
Host on github ONLY after the exam /1

LEVEL 1 - BASIC HTML / CSS

There are only 2 components / <div>s.
- controls
- display

Controls
- menu <button>
- text <input> for backgroundImage filename
- text <input> for typing in a title
- text <input> for typing in a description
- color <input> for selecting a text color
- + <button> for creating new components / <div>s

The control <div> is fixed at the bottom center. Only the menu button is showing, while all other controls are off the screen.

Except for the color <input> and + <button>, every single control is 100% width of the controls width.

Display
- A background <div> that contains both title <div> and description <div>
    - title <div>
    - description <div>

The display <div> is positioned absolutely at the top left with 100vw. The background <div> is 100% of the display <div> and 300px in height. Make sure the position is relative as well because you'll be positioning the title and description <div> inside.
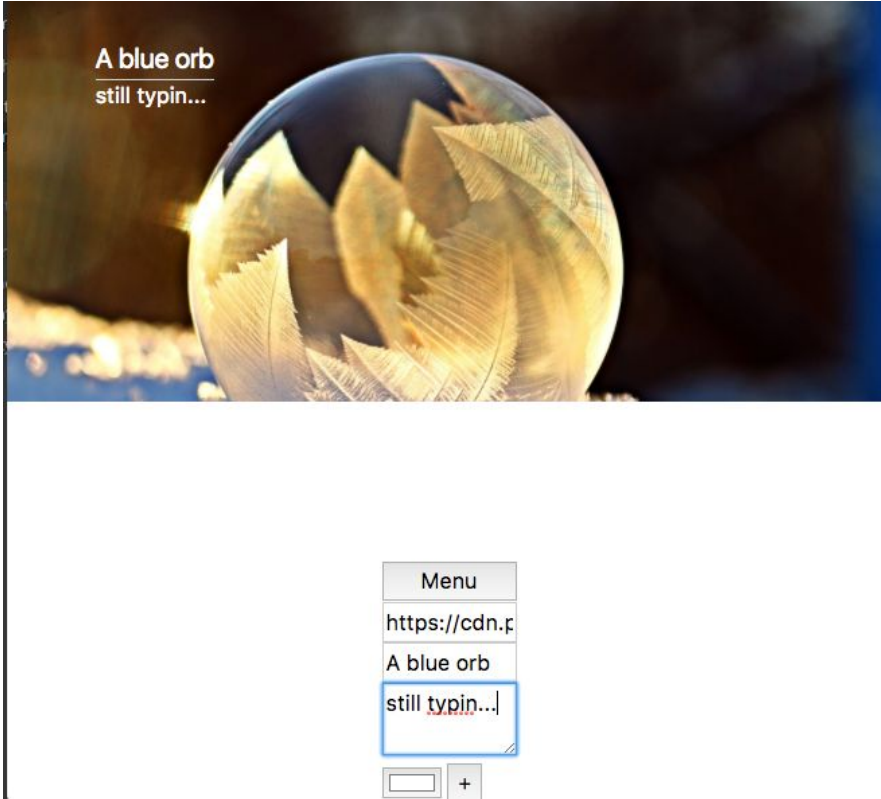
The title and description <div> are both positioned absolutely 10% from the left. The title is 10% from the top while the description is 20% from the top.

The title also has a bigger font-size and a grey border-bottom.

MARKS BREAKDOWN /9
For every incorrect style you lose 0.5-1 mark. 0.5 if you are on the right track. 1 if you are completely off / missing the style.

LEVEL 2 - SIMPLE JS

Interactions
1. Menu expands upward when clicked
2. Pressing return on the background text <input> changes the backgroundImage for the background <div> to the <input> value
3. Every time a key is pressed in the title <input>, the title <div> changes to the <input> value
4. Every time a key is pressed in the description <input>, the description <div> changes to the <input> value
5. Every time a color <input> is changed, change the title <div> and description <div>'s color style to the color <input> value (you will need to use the "change" event here instead of "click" or any of the key)

MARKS BREAKDOWN - /5
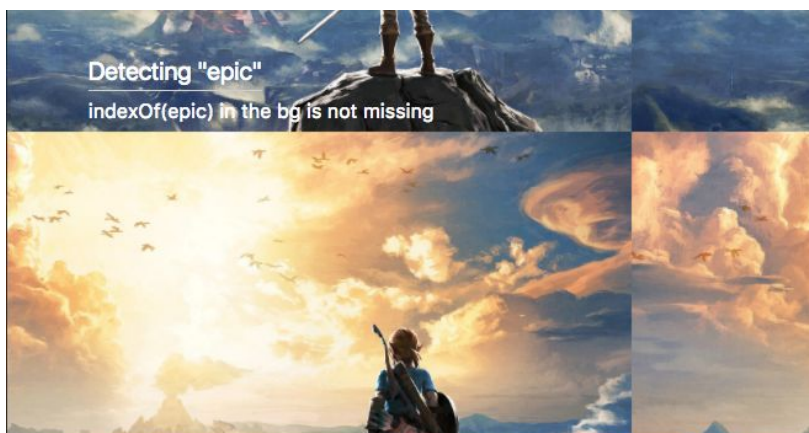
LEVEL 3 - CALCULATIONS
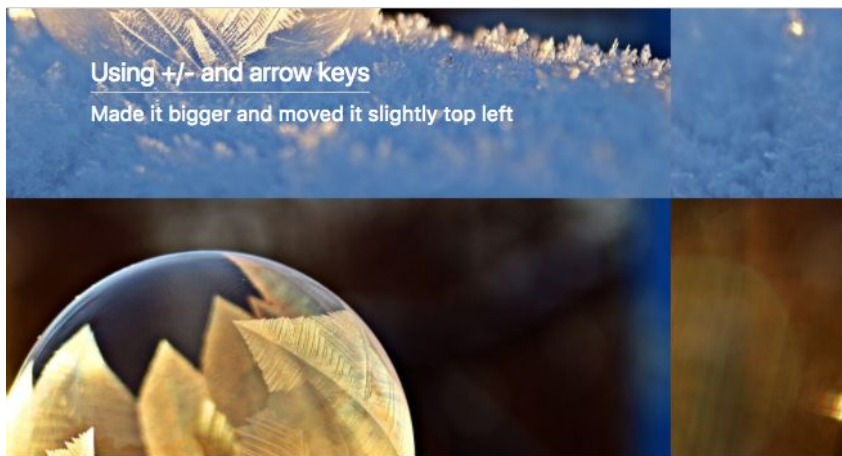
Below interactions all uses the keydown event
Interactions
1. Pressing **DOWN** the right/left/up/down arrow keys moves the backgroundPosition right/left/up/down by 10px
2. Pressing on +/- keys increases and decreases the height of the background <div> by 10px

MARKS BREAKDOWN /3

LEVEL 4 - IF/ELSE

There are several images in the img folder. On top of typing in links, you need to detect the following when changing the backgroundImage. Typing in:

- "horse" gives you img/bg1.jpg
- "night" gives you img/bg2.jpg
- "mountain" gives you img/bg3.jpg
- Using indexOf, if any part of the text contains the word "epic", it'll give you img/bg4.jpg
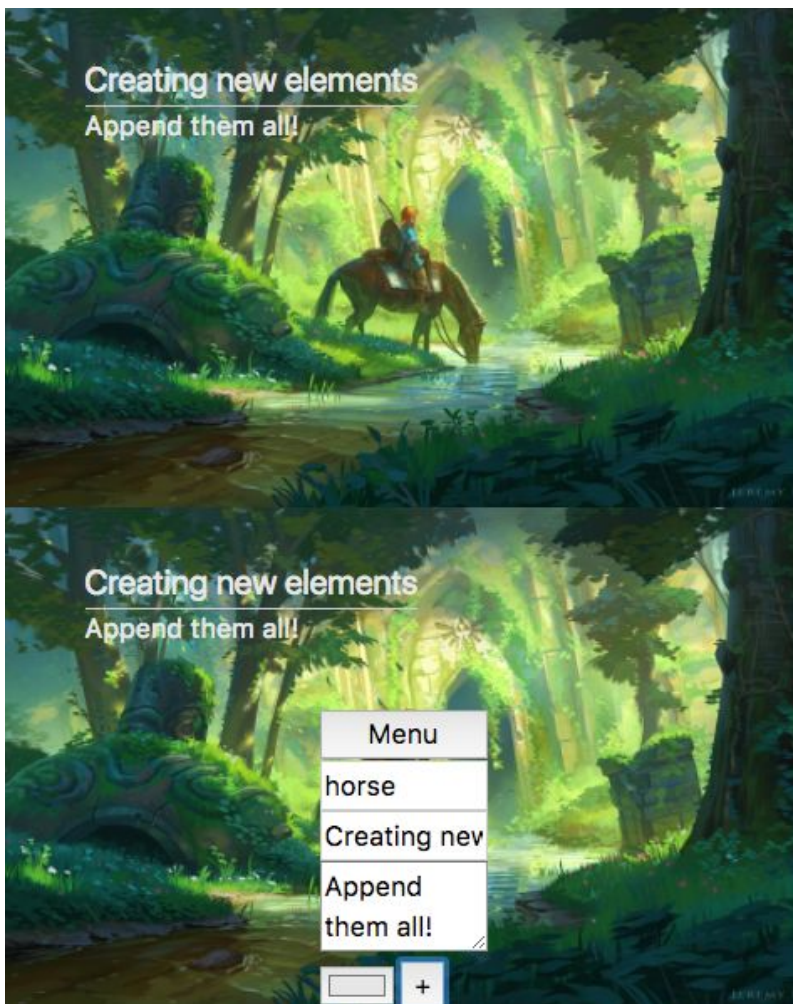
MARKS BREAKDOWN /3

LEVEL 5 - FUNCTIONS

Break everything up into the following functions. Use the functions in the interactions.
- expandMenu()
- changeColor()
- changeDescripton()
- changeTitle()
- changeBG()
- moveBG(keyCode)

MARKS BREAKDOWN /3

LEVEL 6 - CREATING NEW ELEMENTS

We need to be able to create new storyboards with title and description. So create more background <div> with title and description <div> appended. And then append the background <div> into the display <div>.

Use classes (className) to style them all properly.

Assign the newly created <div>'s style with the current background <div>'s, title <div>, and description <div>'s style by using something call style.cssText,

Pass down the innerHTML of the title and description <div>s as well.

PS Make sure you style the controls <div> with a z-index of a huge number so that it stays infront.

MARKS BREAKDOWN /3

LEVEL 6.2 - BOOTSTRAP
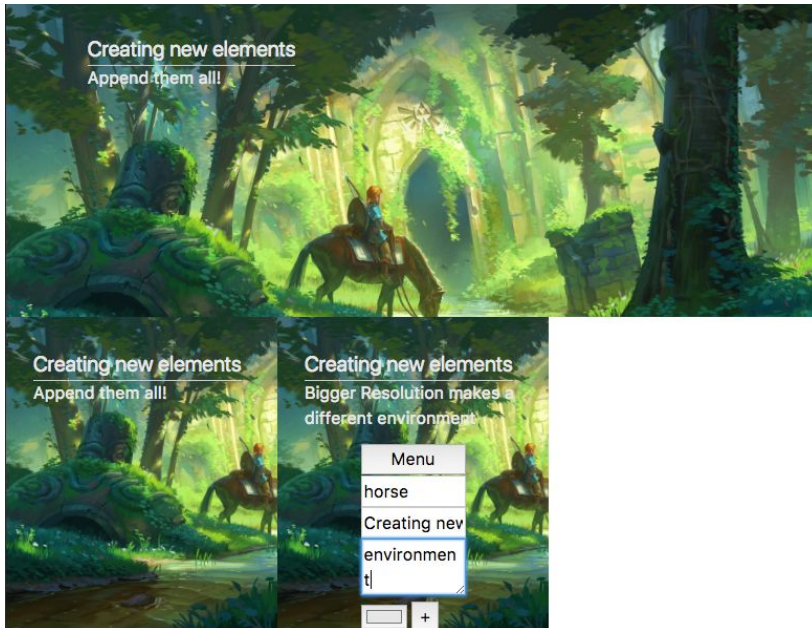
As you create, also use bootstrap's class to make the following responsiveness. For the newly created <div>s :
- Occupy all 12 columns with extra small resolution
- Occupy 6 columns with small resolution
- Occupy 4 columns with medium resolution
- Occupy 3 columns with large resolutions

MARKS BREAKDOWN /2

FINAL CHALLENGE

Make it so that you're always editing the latest newly created <div>s with the controls.

MARKS BREAKDOWN /1

BONUS /1
Make it so that I can click on any of the element and start editing/transforming that specific element (including the ones newly created).

To do the background effectively, you'll have to extract the individual background positions by using a split function.

How split works.
```
var str = "10 15";
var split = str.split(" ");
console.log(str[0]) //outputs 10
console.log(str[1]) //outputs 15
```