In this lab we are going to take what we have learned about both HTML and JavaScript and use it to create an attractive interactive page.

Today we will create a web page that lets you battle against the Thanos. In this game we will be the Avengers consisting of Iron Man, Captain America, Spider Man and Star Lord who are the last defendants against the Thanos. We will be able to select among the four fighters each time to attack. To do this, we will explore the use of JavaScript functions as well as revisit HTML and CSS.

**A. Setting up the HTML**

We’re going to set up our basic page HTML. Most of the tags should be familiar to you, as you have used them in previous labs and homework assignments. If you are having trouble, revisiting these assignments may be useful.

1. Create a new “Lab8” folder inside your CSE3 folder. With Notepad++, create a file named **Avengers.html**
2. Set up the HTML skeleton as we have done in previous labs and homework assignments.
3. Make the title of the page **“Avengers Battle”**.
4. Center the code that will be in the body by putting in `<center>` tags.
5. Create a heading with the `<h1>` tag with the words **“Avengers vs. Thanos”**.

Up until this point, your HTML should look something like this:

```html
<html>
<head>
  <title>Avengers Battle</title>
</head>
<body>
  <center>
    <h1>Avengers Battle: Avengers vs. Thanos</h1>
  </center>
</body>
</html>
```
Our Page is going to have a screen which looks like this:

**Avengers Battle: Avengers vs. Thanos**

The Thanos

HP 1000

Avengers

HP 2000

Which Avenger should attack?

<table>
<thead>
<tr>
<th>Iron Man</th>
<th>Captain America</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP 500</td>
<td>HP 500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spider Man</th>
<th>Star Lord</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP 500</td>
<td>HP 500</td>
</tr>
</tbody>
</table>
Now we are going to create the game screen, starting with displaying the Avengers and the Thanos and their stats:

7. Create `<table>` tags.
8. Create two rows with the `<tr>` tags. We will use the `<td>` tags nested in row tags to create two columns for each row.
   a. The first row is for Thanos, who starts with 1000 health points. The second row will be used for Avengers, who start with 2000 combined health points.
   b. The first row, first column and second row, second column will contain the Hero’s and villain’s name and its current health points (HP). For the HP, use a set of `<p>` tags, each with id attributes. The first id should be `thanosHP` and the second id should be `avengersHP`. This will be explained later.
   c. The remaining cells will be used to display the images. Download the two images linked under Lab 8 on the CSE3 website and save in your Lab8 folder them as “thanos.jpg” and “avengers.jpg”. Use the `<img>` tag to display the images in the correct cells. Give the thanos img id `thanos` and give avengers img id `heroes`.

Your Table in HTML should look as follows:

```html
<table>
  <tr>
    <td>
      The Thanos
      <p id="thanosHP">HP 1000</p>
    </td>
    <td>
      <img src="thanos.jpg" id="thanos"/>
    </td>
  </tr>
  <tr>
    <td>
      <img src="avengers.jpg" id="heroes"/>
    </td>
    <td>
      Avengers
      <p id="avengersHP">HP 2000</p>
    </td>
  </tr>
</table>
```

Next we will create a dialogue box to display the events of our attacks in the battle.
a. Underneath your table, use a set of `<div>` tags and give it an id attribute of `screen`. You do not have to worry about what this tag is or what it does for this class. Just think of it as a container on a webpage.

b. Inside the `<div>` tags, use a set of `<p>` tags with the following text: **Which Avenger should attack?**

```html
<div id="screen">
  <p>Which Avenger should attack?</p>
</div>
```

Finally, we will add the buttons to our game:

1. Create another table with two rows and two columns.
2. To create the buttons, we will use the `<button>` tag to place a button in each of the table cells. The button tag has a special attribute called `onclick`. We will leave this blank for now, but we will use it to add functionality later.
3. In between each of the button tags, we will put the name one of Avengers attacks.
   a. In the top row you should have **Iron Man** and **Captain America**.
   b. The second row should have **Spider Man** and **Star Lord**.
4. Add a `<p>` tag to all the heroes which specify their health points of 500HP
   Eg: Iron man has the id “ironmanHP”

Your second table will now look like this:

```html
<table>
  <tr>
    <td>
      <button onclick="" >Iron Man</button> <p id="ironmanHP">HP 500</p>
    </td>
    <td>
      <button onclick="" >Captain America</button> <p id="captainHP">HP 500</p>
    </td>
  </tr>
  <tr>
    <td>
      <button onclick="" >Spider Man</button> <p id="spidermanHP">HP 500</p>
    </td>
    <td>
      <button onclick="" >Star Lord</button> <p id="starlordHP">HP 500</p>
    </td>
  </tr>
</table>
```
B. Formatting with CSS
As we did before, we will use CSS to efficiently format and style our web page.

1. In between the `<head>` tags, create a set of `<style>` tags as before:

```html
<style type="text/css">

</style>
```

2. We want the `<body>` background color to be **Black** and the font `<color>` to be **white**.
3. We want to use the CSS to set the height and width of our images. Set the `<img>` `<height>` and `<width>` to **200px**.
4. We will also set the `<height>` of the `<div>` to **120px**. Its `<width>` to **400px** and give it a `<border>` that is **2px solid white**.
5. Lastly, we will make `<button>` have a `<height>` of **50px** and `<width>` of **200px**.

It should look like this:

```css
body{
  color: white;
  background-color: black;
}
img{
  height: 200px;
  width: 200px;
}
div{
  height: 120px;
  width: 400px;
  border: 2px solid white;
}
button{
  height: 50px;
  width: 200px;
}
```
C. Our JavaScript functions

Functions are used in many programming languages to encapsulate a set of instructions to be executed. Oftentimes this is code that is used multiple times with varying input. This saves us the trouble of the same set of instructions copied in different places in the program. Instead, we simply perform a function call.

In our Avengers battle game, we will implement 3 functions: (1) `attack()` to attack one round of the battle, (2) `thanosAttack(hero)` to determine what move Thanos will do, (3) `avengersAttack(hero)` to calculate the damage that Avengers’s attack generates.

1. Put `<script>` tags in the `<head>` tags to denote JavaScript code. All our JavaScript code will reside in here:

```html
<script type="text/javascript">
</script>
```

2. Next, we must declare our functions. The syntax looks similar to that of the loops; the code within the curly braces are executed each time the function is called. Functions also have a set of parenthesis. This is used to pass values to the function. For `avengersAttack(hero)` and `thanosAttack(hero)`, we will pass in the selected hero as the parameter to generalize our code for all heroes.

3. For the function `attack(hero)`, we will call our `thanosAttack(hero), avengersAttack(hero)` and `result()` functions. Functions are called by their name, followed by a set of parentheses:

```javascript
function attack(hero){
    thanosAttack(hero);
    avengersAttack(hero);
    result();
}
```

4. Next we will declare and initialize some variables at the top of our JavaScript code to use later in our functions. We will need:

1) `thanosHP` - initialize this to 1000 to keep track of Thanos health points
2) `avengersHP` – initialized to 2000 to keep track of Avengers’s health points
3) `avengersAttackName` – initialized to the empty string. This will be the name of Avengers’s attack.
4) **avengersAttackPower** – initialized to 0. This is the strength of the move avengers’s uses.
5) **thanosAttackRank** – the rank of Thanos attack
6) **thanosAttackName** – initialized to the empty string. The name of Thanos attack.
7) **thanosAttackPower** – initialized to 0. The strength of Thanos attack.
8) **ironmanHP** – Initialized to 500. The health of iron man.
9) **captainHP** – Initialized to 500. The health of Captain America.
10) **spidermanHP** – Initialized to 500. The health of Spiderman.
11) **starlordHP** – Initialized to 500. The health of Star Lord.
12) **play** – Set to true. It is used to check if the player is alive.

5. Now we will write our **thanosAttack(hero)** function.

   a) Use JavaScript’s `Math.random()` function to generate a random number between 0 and 1(exclusive). We will then multiply this number by 2 and add 1 to get a number between 1 and 2. Lastly, we use JavaScript’s `Math.floor()` function, passing in that number as a parameter to round down to the nearest whole number. This number is used to randomly choose Thanos attack. Assign its value to the variable **ThanosAttackRank**:

   ```javascript
   function thanosAttack(hero){
     if(play==true)
     {
       thanosAttackRank = Math.floor((Math.random() * 2) + 1);
     }
   }
   ```

   b) We will have a set of if and else statements to check the value of **thanosAttackRank**. Depending on its value, we will assign the appropriate value of the attack’s strength to **thanosAttackPower** and the corresponding name to **thanosAttackName**.

   c) Based on the attack we change the picture associated in the game. We use the code `document.getElementById("thanos").src` to energy blast and infinity punch.

   d) **Thanos** attacks are listed in the table below:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Energy Blast</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Infinity Punch</td>
<td>150</td>
</tr>
</tbody>
</table>
Your set of if/else statements should look somewhat like:

```javascript
if(thanosAttackRank == 1){
  thanosAttackPower = 200;
  thanosAttackName = "Energy Blast";
  document.getElementById("thanos").src="/thanosblast.jpg"
}
else{
  thanosAttackPower = 150;
  thanosAttackName = "Infinity Punch";
  document.getElementById("thanos").src="/thanospunch.jpg"
}
```

e) We will use a set of if/else statements to decide which hero get attacked which we passed in our function.

We will decrement avenger hero health with the strength of the thanos attack. We will visually change the health of the hero by using the code. JavaScript has a useful command that allows us to modify existing HTML code dynamically:

`document.getElementById('hero'HP). Replacethe hero with the name of the heroes.

In plain English, the command is saying:

a) Look through this document (HTML page).
b) Find the element (tag) in this document by its ID attribute (getElementById).
c) Get the HTML code within this element’s tags (innerHTML).
d) Replace the HTML code within the tag with the new value (the new value here is the string “HP” followed by the value stored in ‘hero’HP).`
Afterwards, we will want to decrement avengers health points by the strength of thanos attack.

```
avengersHP -= thanosAttackPower;
```

6. After the opponent attacks, we will want to visually update Avengers health points and print some dialog onto the screen.

7. Let’s take a look back to the table HTML and find the `<p>` tags that we defined earlier. The last line of code we wrote will insert the value stored in `avengersHP` in between the `<p>` tags with the id attribute `avengersHP`. That line of code says to look through the document and get the tag with the id called `avengersHP`, which is the first cell in the second column of our first table. Now innerHTML will get the text in between the opening and closing `<p>` tag. We then set the value of “`HP` + `avengersHP` as the text in between the opening and closing tag.

Similarly, we will want to use our `document.getElementById().innerHTML` call to look for the tag that has the id attribute of `screen`. We will replace the HTML in there with a message stating which attack Thanos used and how much health avengers lost. Make sure to wrap this information in a set of `<p>` tags.

```
document.getElementById("avengersHP").innerHTML = "HP " + avengersHP;
document.getElementById("screen").innerHTML = "<p>Thanos used " + thanosAttackName + "! Avengers lost " + thanosAttackPower + "HP.</p>";
```

That’s it for the `thanosAttack(hero)` function.

8. Now let’s go back to the HTML code for our Avengers attack buttons. The value assigned to the `onclick` attribute is javascript code that will be executed. In between the quotations for the `onclick`
attribute, put `attack('hero name')`. This is a function call, and it will call the `attack('hero name')` function to execute its code.

```html
<button onclick="attack('Iron Man')">Iron Man <p id="ironmanHP">HP 500</p></button>
```

Now if you run the code and click any of avenger member buttons, you should see the dialog box display one of thanos attacks. You should also notice that avengers HP and the hero hp has decreased.

9. Now let’s let the Avengers retaliate in battle. In our `avengersAttack(hero)` function, we will write code for avengers attack functionality.

   a) We want to choose which avengers member will attack based on which button we press. When the user clicks on one of the buttons, we will pass the appropriate hero to our `avengersAttack` function. Inside or your attack buttons’ `onclick` attribute, we write code to pass in the parameters for the function.

Each avengers Members attack stats are as follows:

<table>
<thead>
<tr>
<th>Avengers Member</th>
<th>Attack</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iron Man</td>
<td>Laser Blast</td>
<td>100</td>
</tr>
<tr>
<td>Captain America</td>
<td>Shield Punch</td>
<td>150</td>
</tr>
<tr>
<td>Spider Man</td>
<td>Web Shoot</td>
<td>80</td>
</tr>
<tr>
<td>Star Lord</td>
<td>Gun Barrage</td>
<td>90</td>
</tr>
</tbody>
</table>

b) Inside of the body of your `avengersAttack()` function, similarly to your `thanosAttack()` function, put in some conditional statements to check for the name of the hero and assign the correct name and strength to the variables `avengerAttackName`, `avengerName` and `avengerAttackPower`. We also check the HP for the hero. We use an if/else condition to check if the hero’s HP is above 0. If it is we set the play boolean to true else we set it to false. In the If condition, we change the avengers image to the image of the hero. We use the `document.getElementById("heroes").src` to change the image to the hero we need. In the Else statement, we set the play Boolean to `false`. We use `document.getElementById("screen").innerHTML` to set the dialogue to the hero being dead. We use the key word `return;` to go exit the function as we do not wish to continue the attack.
Continue to repeat the same format for all four Avengers members.

c) Next, decrement thanosHP by the strength of avengers attack.
d) Update the display of information by using `document.getElementById().innerHTML` to change the HTML inside of the tag with the id `thanosHP` to display Thanos’s new health points.
e) Lastly, we will want to add text to our dialog box to display which Avenger Member attacked and what power it used and how much health Thanos is lost. This time we will use the `+=` operator to append the information rather than replace it.

```
document.getElementById("screen").innerHTML += "<p">hero</p> used " + avengersAttackName + "!
Thanos lost " + avengersAttackPower + " HP.";
```

Now when you run your webpage, both Avengers and Thanos should be able to attack.

Now we going to have the `result()` function to let us know who won at the end of battle.

1. We will start by declaring the function `result()`.
2. We use `if` and `if-else` to check who won the war by checking if `thanosHP` is less than 0 and else if `avengersHP` is less than zero.
3. Update the display of information by using `document.getElementById().innerHTML` to change the HTML inside to display the result.

```
function result(){
  if(thanosHP<0)
  {
    document.getElementById("screen").innerHTML = "<p">Avengers have saved the world</p>";
  }
  else if(avengersHP<0)
  {
    document.getElementById("screen").innerHTML = "<p">Thanos has destroyed the world</p>";
  }
}
```
D. Putting it all online

1. Modify your CSE3Page.html to include a link to your avengers.html.
2. Use Secure File Transfer Client to update your CSE3 folder online.
   REMEMBER to drag ONLY your Lab8 folder into your CSE3 folder

Check-off:
- Updated CSE3Page.html online with Lab8
- Working avengers game