NM2207

Session 06

Assignment

Due: 28th September 11.59 pm

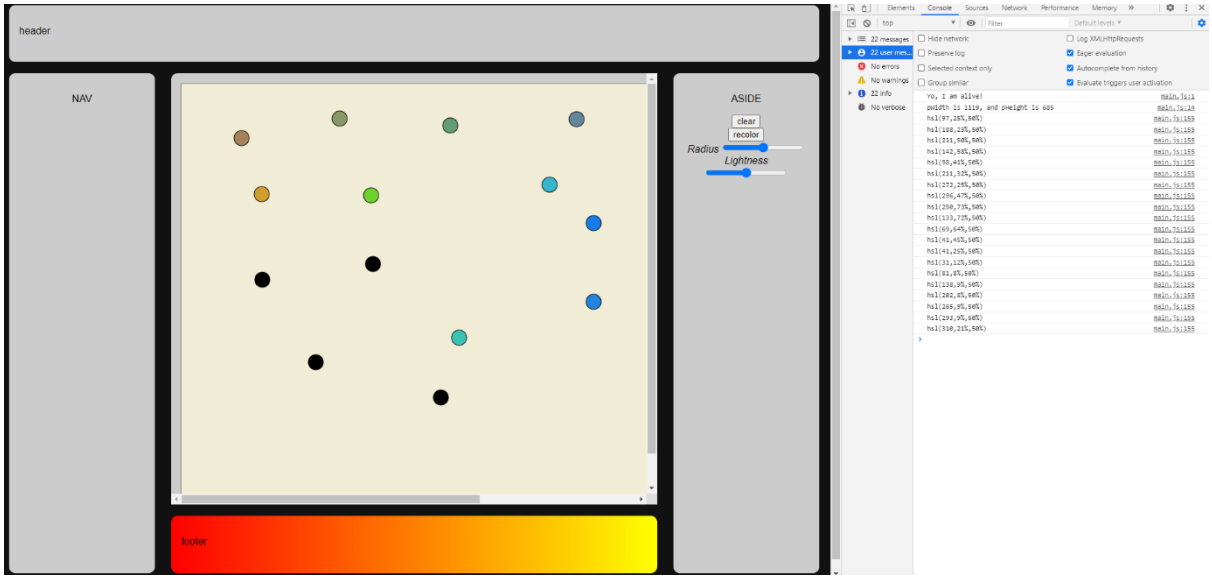
Overview of what we'll do today:
Return to what you created in Session05.class. Each time a circle is created, it gets a new color, but the color is not random. To clarify:

- The lightness is set by a new slider.
- The x coordinate of the circle determines the hue
- The y coordinate of the circle determines the saturation

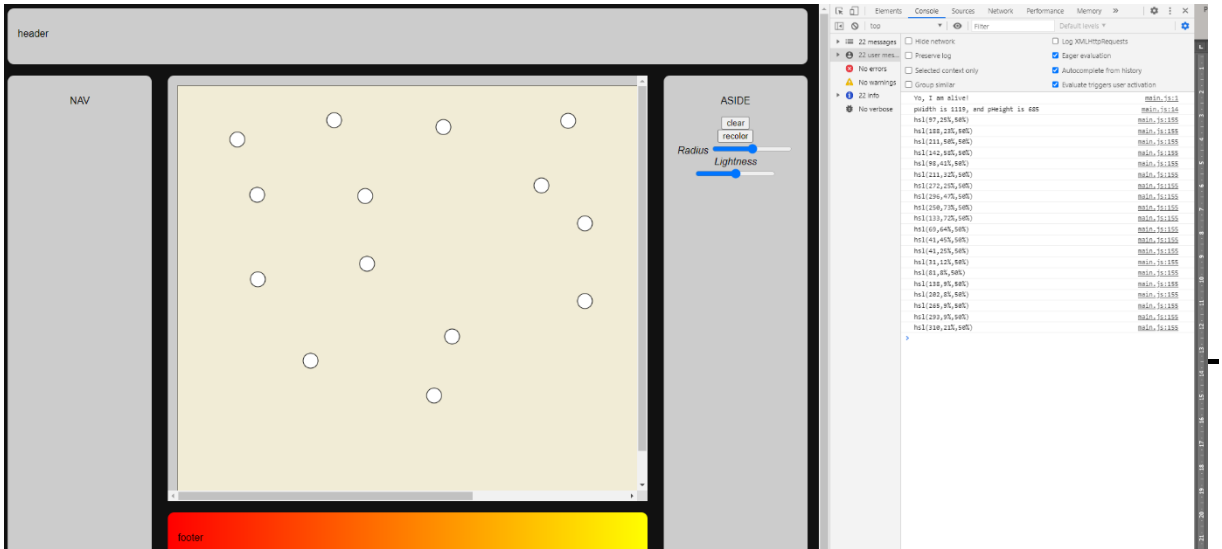
Main challenges:

1. To set the hue and saturation each time a circle is created, using the properties of the event variable
2. To set the lightness using a second slider in the aside element.
3. To map them into the right range using the map function
4. To wipe out the colors later by using DOM collections

When you are clicking around and creating circles:



After recolor button is clicked:



Part 1

Create and use a new slider value as the lightness component of the hsl() color string.

1.1 A new slider and a new button have already been added into the view.

1.2 Add an event listener to the slider which updates a global variable called lightnessValue each time the slider value is changed.

1.3 When the button is pressed, it should also trigger the changeColor function which takes in no arguments and returns no values. This function is already provided in the template.

1.4 Print a message to the console.log whenever the button is clicked.

Part 2

Define your variables and create new colors

Lets represent the color of a circle as an object with three properties, hue, saturation and lightness.

2.1 Now we will edit the drawCircle function. When a circle is created, also create an HSL value and use it as the fill attribute for the circle.

2.2 Define a new object as

```
var circleColor = {};
```

2.3 We will give circleColor three properties.

- So circleColor.hue would be the cx coordinate of the circle, or the ev.offsetX.
- And circleColor.saturation would be the cy coordinate of the circle, or the ev.offsetY.
- Finally, circleColor.lightness would be the value of the lightness slider.

2.4 Each time a circle is created in the drawCircle function, format a new HSL string by making a call to your hslString function (available in the template). The input argument will be your circleColor object. And it will return a formatted HSL string.

2.5 So obviously, the major difference is that now we will pass the hslString function an object and not three arguments.

2.6 But what is the range of ev.offsetX and ev.offsetY? What is the range of the lightness slider? How do they compare to the range of hue, saturation, and lightness values? How shall we fix this?

Part 3

Map your HSL attributes into new ranges

3.1 Before formatting an HSL string, you will first need to map each of your hue, saturation and lightness values into the right range. Map each of them them into the correct range by calling the map function in your drawCircle function.

3.2 Try to work this out: what are x, a,b,m, and n arguments? What is the output of this function?

- a and b are the minimum and maximum values that a can take. E.g., ev.offSetX can take a minimum value of 0 and a maximum value of pWidth or dimX.
- m and n are the minimum and maximum values that hue, saturation and lightness can take

3.3 Use the returned value to set the new color of the circle.

Part 4

Wipe out the colors!

Edit your changeColor function to collect and store the data you need for this problem: a list of circles!

Make the following changes to your currently empty changeColor function:

3.4 First, this function will create a list of circles by using document.getElementsByTagName.

3.5 Now, you can access the fill color of each circle by using something like this:

```
for (x in circleList) //execute the statements in curly braces for each circle in the list.
{
        var filledValue = circleList[i].style.fill; //use this to GET the color
        console.log(i); //what is i??
        circleList[i].style.fill = ??; // use this to SET a new fill color
}
```