Web Programing JavaScript – Bonus Bonus Task
Graphics 11/18/2023
2 - Parts

Please see the Graphics class notes for more help- Below are the sample code

Html Content

```
<div>
  <label for="xSlider" style="width:100px; display:inline-block">
    Scale x: <b id="xScale">1</b>
  </label>
  <input id="xSlider" type="range" min="0" max="3" step="0.1" value="1">
</div>
<div>
  <label for="ySlider" style="width:100px; display:inline-block">
    Scale y: <b id="yScale">1</b>
  </label>
  <input id="ySlider" type="range" min="0" max="3" step="0.1" value="1">
</div>

<canvas id="myCanvas" width="400" height="200" style="border: 1px solid gray"></canvas>
```

JavaScript Content

```
// Redraw the sun when the x slider value changes
let xScaleElem = document.getElementById("xScale");
let xSlider = document.getElementById("xSlider");
xSlider.addEventListener("input", function() {
  xScaleElem.innerHTML = xSlider.value;
  drawSun();
});

// Redraw the sun when the y slider value changes
let yScaleElem = document.getElementById("yScale");
let ySlider = document.getElementById("ySlider");
ySlider.addEventListener("input", function() {
  yScaleElem.innerHTML = ySlider.value;
  drawSun();
});

let canvas = document.getElementById("myCanvas");
let context = canvas.getContext("2d");

drawSun();


function drawSun() {
  // Clear context of previous image
  context.clearRect(0, 0, canvas.width, canvas.height);

  // Save current context settings
  context.save();

  context.beginPath();
```

```
 // Scale sun based on slider values
 context.scale(xSlider.value, ySlider.value);

 // Draw sunbeams (4 rotated squares)
 for (let i = 0; i < 4; i++) {
  context.translate(70, 70);
  context.rotate(Math.PI / 8);
  context.translate(-70, -70);

  context.fillStyle = "orange";
  context.fillRect(20, 20, 100, 100);
 }

 // Draw interior circle
 context.arc(70, 70, 50, 0, 2 * Math.PI);
 context.fillStyle = "yellow";
 context.fill();

 // Restore previously saved settings
 context.restore();
}
```

The save() and restore() methods are useful when a developer wants to undo transformations that have been applied to the context.

The sun is currently scaling out from the origin at (0, 0). Make the following modifications so the sun appears in the center of the canvas and scales about the sun's center:

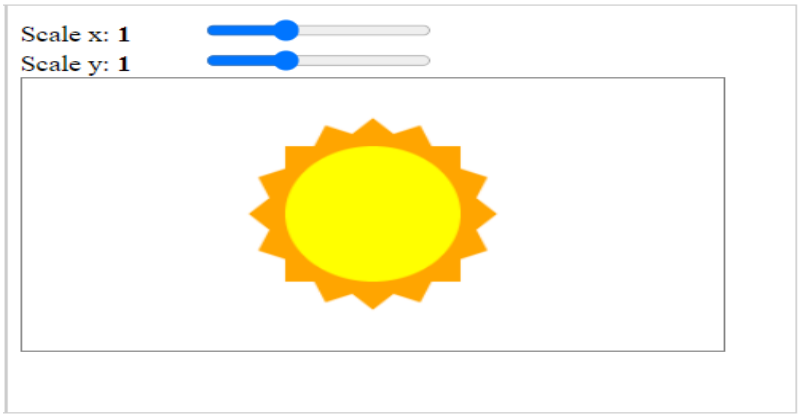
Part - 1

What to do ?

1. Call context.translate(130, 30) before scale() so the origin is at (130, 30), and the sun appears in the center of the canvas. Press "Render web page" and note that the sun is now initially centered but scales as before.

2. Call context.translate(70, 70) before scale() so the origin is in the center of the sun before scaling. Then, the call to scale() will cause the sun to scale about the center of the sun instead of the upper-left corner of the sun. Call context.translate(-70, -70) immediately after scale() to move the origin back to the upper-left corner of the sun. Press "Render web page" and note that the sun is now scaling from the center of the sun.

Expected webpage



Part - 2 -

What to do ?

The web page below shows a basketball image that bounces left and right when the user mouses over the canvas. The basketball logic is encapsulated in a ball object that has 2 methods:

- move() - Moves the x coordinate of the ball left or right and accounts for bouncing the ball off the left and right sides of the canvas
- draw() - Draws the ball at the ball's x and y location on the canvas

The ball's methods are called in drawFrame(), the callback function supplied to requestAnimationFrame(). When the mouse is moved off the canvas, the cancelAnimationFrame() method is called, which cancels the animation request.

Make the following modifications so the ball bounces off all walls, rotates, and scales larger:

1. Add code to the ball's move() method so the y property is incremented/decremented just like the ball's x property. Add an if statement to multiply yInc by -1 if the y property causes the ball to be drawn past the top or bottom edges of the canvas.

2. Add code to the ball's draw() method to rotate the ball by adding 0.01 to the ball's rotation property. Then, translate to the ball's center, rotate the canvas by rotation, and translate back before drawing the ball. The ball's center can be calculated as:

   centerX: this.x + this.img.width / 2;
   centerY: this.y + this.img.height / 2;

3. Make the ball scale up in size by 0.1 when the ball hits a canvas edge. Modify the ball's scale property when the ball bounces. The ball should scale about the ball's center, so place the call to scale() near the code to rotate the ball.

4. Finally give the scale() a end point of something reasonable.

Here is the code you are working with below..

Html Content

```
<canvas id="myCanvas" width="610" height="370" style="border: 1px solid gray"></canvas>

<div style="display:none;"> <!—you can find any or use an image resource/link that is similar to what I used-->
  <img src="https://c.tenor.com/Forg6sufPCMAAAAM/spinnig-on-fire.gif" id="basketball" alt="basketball">
</div>
```

JavaScript Content

```
let ball = {
  x: 10,
  y: 10,
  xInc: 3,
  yInc: 3,
  scale: 1,
  img: document.getElementById("basketball"),
  rotation: 0,

  // Draw the ball
  draw: function() {
    context.drawImage(this.img, this.x, this.y);
  },

  // Move the ball
  move: function() {
    this.x += this.xInc;

    // Bounce of the left and right canvas edges
    if (this.x < 0 || this.x + this.img.width > canvas.width) {
      this.xInc *= -1;
    }}};

let canvas = document.getElementById("myCanvas");
let context = canvas.getContext("2d");

// Draw ball at starting position
context.save();
ball.draw();
context.restore();

// Used to cancel animation
let animFrameId;

// Start the animation when the mouse is on the canvas
canvas.addEventListener("mouseover", function(e) {
  animFrameId = window.requestAnimationFrame(drawFrame);
});

// Stop the animation when the mouse is moved off the canvas
canvas.addEventListener("mouseout", function(e) {
  window.cancelAnimationFrame(animFrameId);
```

```
});


// Draw a single frame
function drawFrame() {
  context.clearRect(0, 0, canvas.width, canvas.height);

  context.save();
  ball.draw();
  ball.move();
  context.restore();

  animFrameId = window.requestAnimationFrame(drawFrame);
}
```