Properly Resizing your Canvas

The fix for this is pretty straightforward. Don't use CSS for specifying your canvas's size. Instead, set the `width` and `height` attribute values on your canvas element directly:

```html
<!DOCTYPE html>
<html>

<head>
  <style>
    #myCanvas {
      border-width: 1px;
      border-style: solid;
      border-color: Black;
    }
  </style>
</head>

<body>
  <div id="container">
    <canvas id="myCanvas" width=350 height=250>

    </canvas>
  </div>

  <script>
    // omitted!
  </script>
</body>

</html>
```
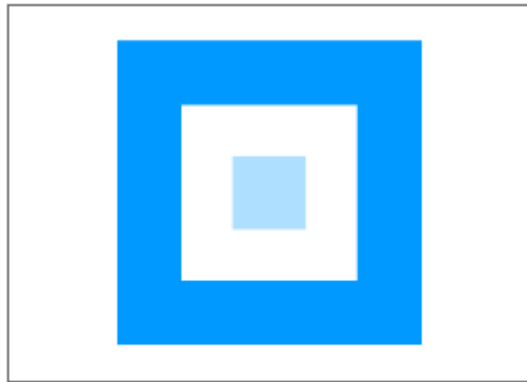
Once you do this, both your canvas as well as the contents inside it will be sized properly:

[happy days are back again!]

A lot of times, the size of your canvas element will not be fixed. You may want to adjust its size based on some interaction, resize operation, etc. For these cases, you will need to set your `canvas`'s size programmatically.

The following code snippet explains how you can do that:

```
var myCanvas = document.querySelector("#myCanvas");
myCanvas.width = 350;
myCanvas.height = 250;
```

I am basically getting a reference to our canvas element and setting the width and height attributes. This is basically the JavaScript version of the HTML that I modified earlier.

If you want to run your canvas in fullscreen mode and ensure it always takes up all of the available space, the code for making sure it is sized (and resized) correctly is:

```
window.addEventListener("resize", resizeCanvas, false);

function resizeCanvas(e) {
  var myCanvas = document.getElementById("myCanvas");
  myCanvas.width = document.documentElement.clientWidth;
  myCanvas.height = document.documentElement.clientHeight;
}
```

You can learn more about how to calculate your viewport size by looking at the Viewport, Device, and Document Size tutorial.