**Lab 1-5 Observations**

When moving the width and height attributes from the canvas element in index.html to the #game-canvas rule in the CSS file, the webpage changes. While the canvas area appears to be the same as before, the canvas is zoomed in. It zoomed in to the top left corner of the regular canvas, so we can only see the top left corner of the background.

The reason this happens is because the width and height attributes of the canvas element are not the same as the width and height properties in CSS. When dealing with a canvas element, the width and height elements determine the number of pixels that the canvas itself has. In the case of 800px x 400px, that gives the canvas a total of 320,000 pixels. However, if the width and height elements are not specified in the canvas element, it will default to 300px x 150px for a total of 45,000 pixels. This is significantly less than the correct number of pixels, which explains why everything looks “zoomed in.” It’s not actually zoomed, but rather the image is being drawn at full size in just a fraction of the intended pixels. It gets cropped as a result. The CSS width and height properties do not actually determine the number of pixels the canvas has, but what size the browser should display the element as. So, it takes the 300px x 150px canvas and scales it up to 800px x 400px by stretching the image. The CSS rule doesn’t give the canvas more pixels to work with. You can see exactly what is going on by removing the width and height CSS properties without restoring the width and height elements of the canvas. The canvas appears at its true, pixel perfect size.