Jake Hasson

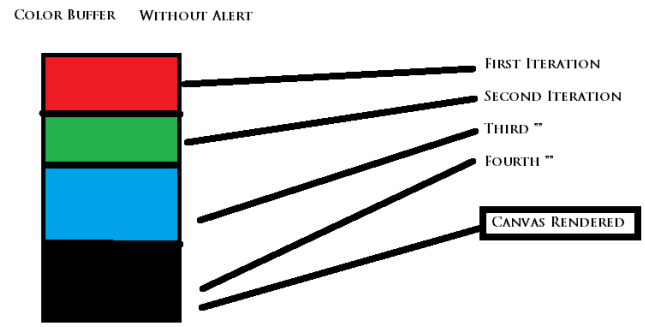
1.) I expected to see a program similar to the one we first implemented where we would see a delay between the drawing of 16 rectangles in a matrix.

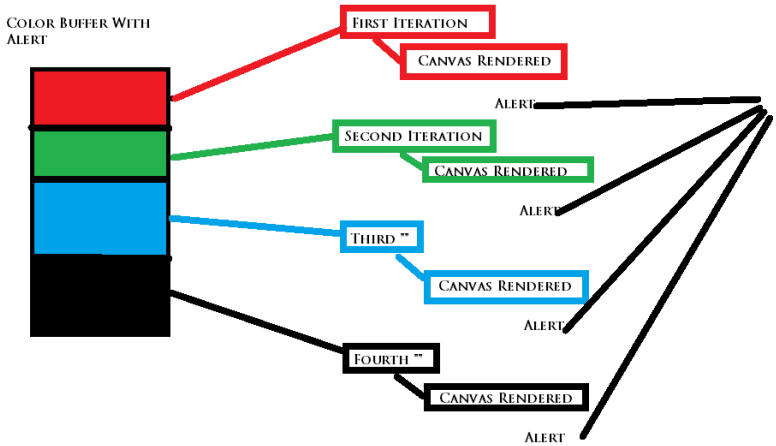
2.) I saw a black rectangle, and upon further inspection(in the debugger by placing a break statement). I saw 4 different colors canvas'. The delay wasn't enough to allow the different colors to be viewed.

3.) Upon changing the alert value to true, we were able to see the difference in color values. The alert placed a "halt" within the for loop that provided the canvas' colors to render in between the changes in color.

4.) WebGL Only supports one color buffer. In the first case, even though there is a delay, this does not change the fact that the canvas will only render upon finishing its main function… as a result… the canvas isn't drawn in the browser until it has been set to black… no matter how long you set the delay for… it will always render black. The last example is different in the sense that because of the "alert()" function, which prompts for an action from the user. It allows the screen to render the different color values for the canvas, because an action is required. It interrupts the main function and "flushes" the color buffer, where it will then be overwriting in the next pass of the for loop until the end of the function.

Diagram





Allow the canvas to be rendered during the for loop prior to the return of the main function because of the "halt" that occurs as a result of the Alert