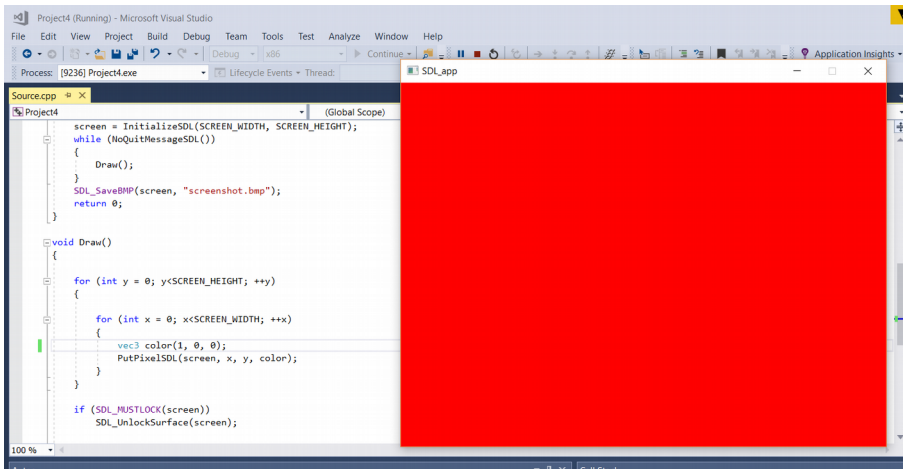
Computer Graphics & Interaction
Lab Assignment #1

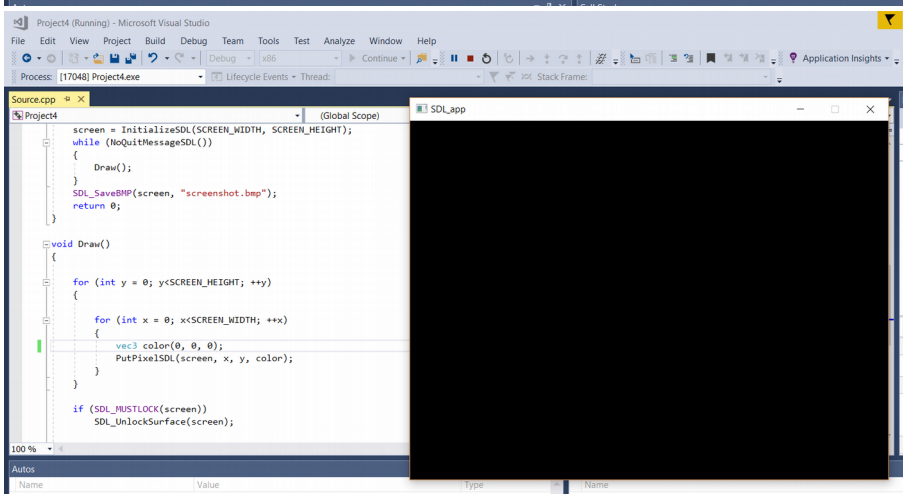
Enrique Perez Soler

2 Intro to 2D Computer Graphics

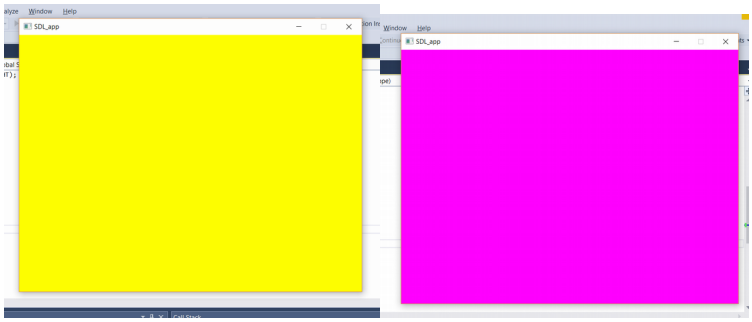
2.1 Color de Screen

Being the first task of the lab to experiment with coloring the screen with different colors, here i present you the samples of the requested colors as well as the tampering with the code. This task i found easy and quite enlightening as it showed how the *PutPixelSDL()* and the *Draw()* functions worked.

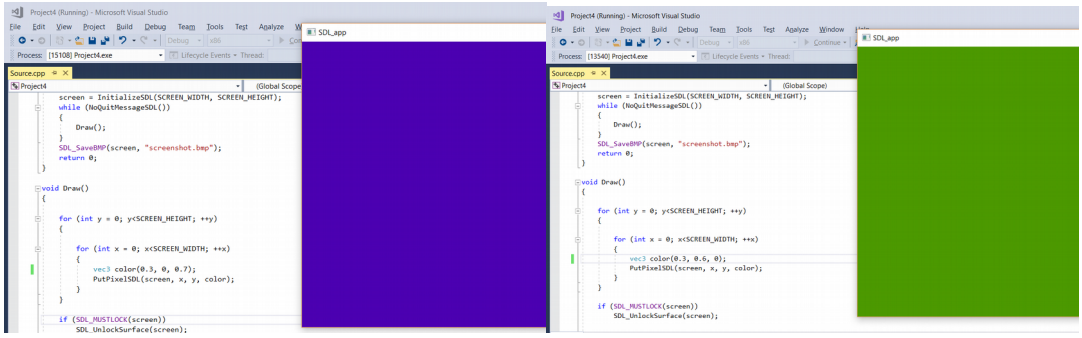




Following the very same procedure I managed to create the other colors:



And here are some colors created by playing with different intensities:

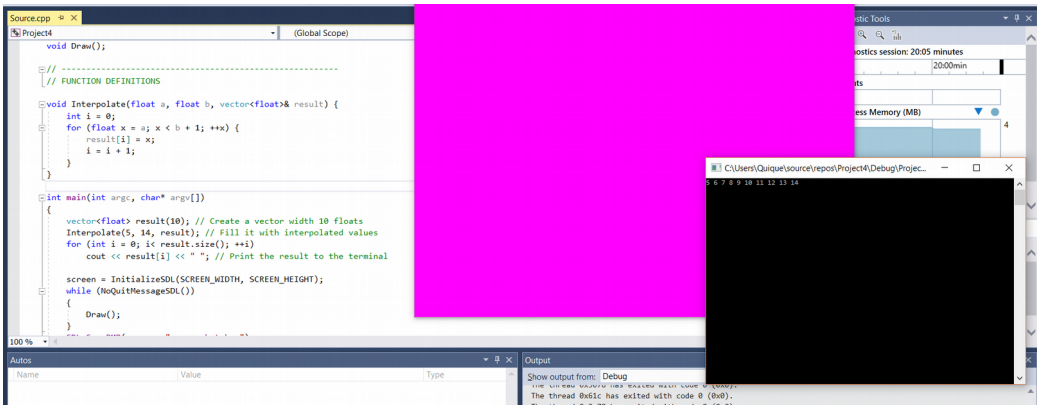


As shown, the different modifications are made in the vector *vec3* where each position represents the respective colors red, green and blue based on the RGB representation.

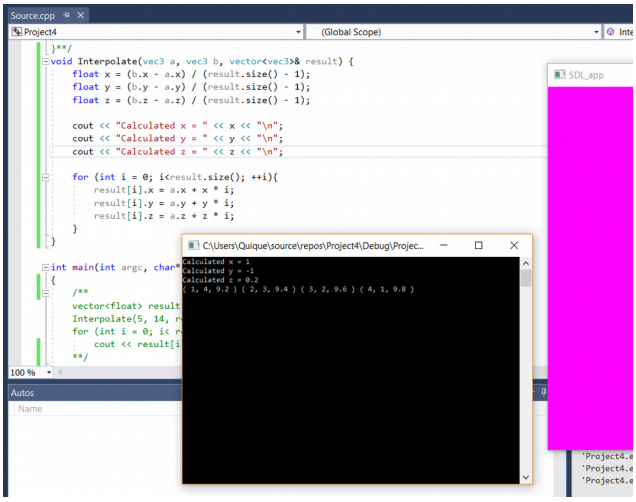
2.2 Linear Interpolation

Taking into account how linear interpolation works basing ourselves in the formula, it was a pretty straightforward implementation of the Interpolation function:

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0},$$

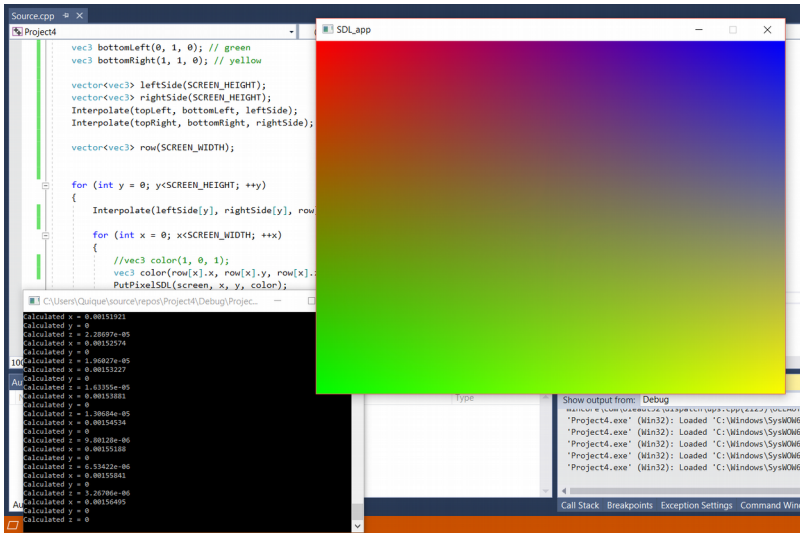


The output in the terminal is as expected so we follow through with the Vector version of the Interpolation which followed the same principle:



2.3 Bilinear Interpolation of colors

Following the steps given in the lab documentation, I managed to use the learned Interpolation technique to adjust to the new use in the Bilinear Interpolation. When executing the result this is what it returned:



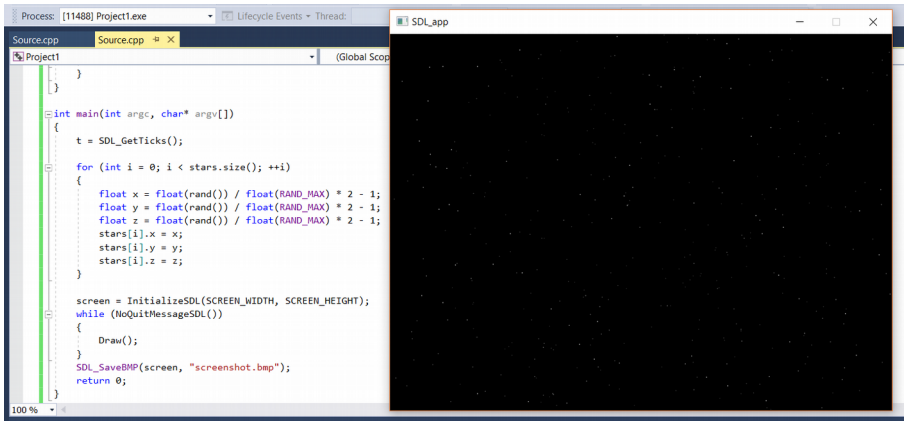
As you can see in the terminal output, I printed the calculations to check for errors and also to have a closer look at how the functions runs through every pixel to color it according to the bilinear interpolation technique.

3 Starfield

I was really excited to start this part when i first read the lab assignment. I have heard about it but i never had the chance to create one so i was pretty receptive and excited. First step was to create a vector that would contain all the stars and in the main() function create a loop that would give them position coordinates (x,y,z) completely randomized.

3.1 Projection pinhole camera

Following the simple steps given I managed to make the right modifications to the *Draw()* function by representing the formula given in the documentation. Here's a snapshot of the resulting starfield:



3.2 Motion

Following the steps given in this section, i took it upon myself to create the new function *Update(),* that will give motion to the stars by being called repeatedly. The stars are not really moving out of the scene but rather creating that illusion by checking every time the position of a star is updated. When and if the star's *Z-coordinate < 0* the position is changed to 1 so it situates the star at the back. Nevertheless, smoothing that change is necessary as when it happens is very notorious that a star is being relocated and thus breaking that illusion of *space travel*. We are gonna achieve this by playing with the intensity of the color of the stars: when they are at the back the intensity is lower than when they are at the front.

Summary & Thoughts

Setting the lab was really the real pain. I have never used Visual Studio but fortunately the guides at the course page helped me through it. Nevertheless, at the time of executing the linear interpolation the terminal didn't pop up. And after a whole day of figuring what went wrong, it turned out the problem was at the "Setting the environment part" that when i created the project it was under the "empty project"(where the terminal is nowhere to be found) and not the "Console Application", where it utilises the terminal as output. As the project was already built i managed to change the subsystem and everything worked just fine.
As for the Starfield task, It took me some time to understand how to calculate the positions of the stars and how to move them as its been a while since i have reviewed my algebra knowledge of vectors and matrices and also I was not that experienced in C++, so some concepts I had to comprehend before i could move forward with the implementations. But overall, this lab proved to be useful at the time of understanding the intricacies of the color rendering. I can't wait to move on to the next lab to keep learning more about it.