

Tutorial 1 - Introduction to C++

Computer Graphics

Kartic Subr and Martin Asenov
September 20, 2019

Part 1: *Hello World!*

Make sure that you have a C++ compiler installed `g++` installed (that should be available out of the box in Linux, Mac and DICE machines). Run a program that prints 'Hello World' through the terminal. Extend the program to collect your name from the terminal and print your name as well. If you are not familiar with C++ you can follow the step here.

Part 2: *Pointers and memory*

An important feature of C++ is the ability for fine grain memory management. As such an important concept is the addresses of in memory where data is stored - *pointers*. This in C++ is primarily controlled via **Address-of operator (&)** and **Dereference operator (*)**. The reason why this is important is when it comes to optimizations of different operations. For example when passing information to a function, one can pass the *address* of the data, rather than copy the data instead. You can read more about the topic here. Make sure you understand `pointers.cpp` and `pointers2.cpp`. Try experimenting with the two operators(& and *) and make sure you understand, when are the two operators used.

Part 3: *Colors!*

You have just started your new job and have been given a task to develop a color calibration sheet, with a set of specific colors. Unfortunately, you have only been given a rough idea of how the sheet should look like and due to some data corruption the colors are in random order.

1. You have been given a `colors.txt` that contains unordered colors in the following format: N, R, G, B , where N is the correct ordering of the color, and R, G, B is the Red, Green and Blue value of the color. Create a 'color calibration sheet'. You can be supplied `SortImage.cpp` that you should extend.
 - (a) You have been supplied code that reads `colors.txt` and as a vector of custom RGB structure, that contains the R,G,B color values plus the N value described above.
 - (b) Visualize all the colors as increasingly big squares on a blank image. Ignore the N values for now, you only need the R,G,B. You should get an image similar to fig.1. (Drawing rectangles using OpenCV)
 - (c) Sort the colors (you can use BubbleSort) based on the first column and visualize again. You should get a image similar to fig.2.
 - (d) Evaluate the execution time against an implementation from the `algorithm` library (found in the code). Do you see a big speed improvement? Why? Why not? What if you used a bigger file to sort - `colors_BIG.txt`.

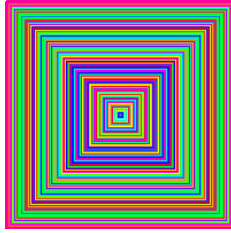


Figure 1: Random colors

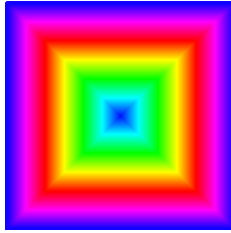


Figure 2: Calibration sheet

Development environment notes. One way to develop C++ would be to use the terminal and a text editor. However, as you start working on your assignments, you might find it beneficial to switch to an IDE - Integrated Development Environment. This will make compiling, linking and **debugging** much easier! I would recommend using MS Visual Studio Code, as it is free, has most of the features you would need, it is available on DICE, and supports for Linux, Mac and Windows (but of course feel free to use any other development environment of choice). You can find some basic instruction on how to setup, debug, etc. here - the instruction here are for Windows, but you can follow a similar setup for Linux/Mac.