Andreas Wrife, David Åstrand, Léo Noharet
awrife@kth.se, dastrand@kth.se, lnoharet@kth.se

DH2323 LAB 1

1. Setup of the Lab Environment

Installing SDL and making the lab environment proved to be the hardest obstacle to overcome in this lab. Since all three group members had different versions of Visual Studio as well as different operating systems, we all encountered different types of errors and found it difficult to help each other. After trying several times with both SDL 1.2 and SDL 2, and many consultations with TAs, we stopped trying to get the lab environment working on our personal computers and opted for completing the lab on the KTH lab computers. After installing the lab code from canvas and running "cmake ." and "make" we were finally able to start working on the lab.

2. Introduction to 2D Computer Graphics

2.1 Color the Screen

After getting used to the lab environment we reverse engineered the code and looked into the draw function. It was obvious how the blue screen was drawn and therefor easy to manipulate by changing the values around following a RGB colour theory. See figure 2.

Figure 1: Different colours obtained by changing the values in the Draw() function.

2.2 Linear Interpolation

To make the rainbow effect, see Figure 3, an interpolation function was implemented where with the help of two input colours a range of the colours in between was created. With the help of the variable $t \in [0, 1]$ the equation used to obtain an interpolation between colour a and b was as follows

$$c = (1 - t) \cdot a + b \cdot t; \tag{1}$$

Since each colour was in the RGB scale, they were represented as vectors with dimension 3 using values between [1, 255].

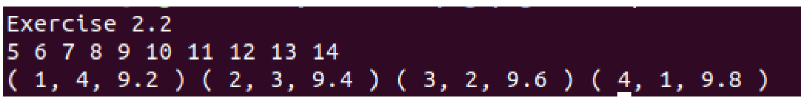

Figure 2: Output from terminal following exercise 2.2, interpolating scalars and vectors.

2.3 Bilinear Interpolation of Colors

A conundrum which was easily solved but seemed daunting at first was how to get the 2D plane of interpolated colours. However, this was solved by simply interpolating between the four corners obtaining two opposite parallel side-vectors and then interpolating between them, obtaining a matrix of interpolated colours.

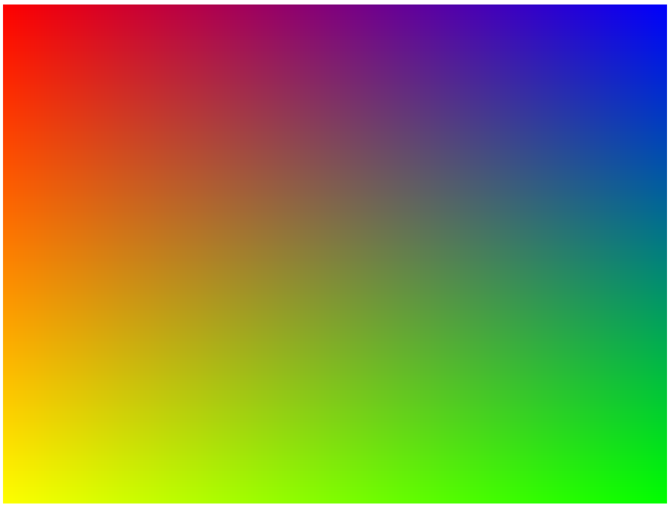

Figure 3: Interpolation between colours Red, Blue, Yellow and Green in the respective corners.

3. Starfield

3.1 Projection - Pinhole Camera

The code for this task can be found in "skeleton2.cpp". To build to the executable for this exercise, it is required to add "skeleton2.cpp" as an executable in the CMakeLists.txt file.

Overall, we encountered little in terms of obstacles while completing this task. The u^i and v^i values were relatively simple to code based on the instructions. Assuming that the screen is 640 wide and 480 pixels tall, the value of $f = \frac{H}{2} = \frac{480}{2} = 240$. The formulas for the field of view are:

$$fov_v = 2 \cdot \arctan(\frac{H}{2} \cdot f) = 2 \cdot \arctan(\frac{480}{2} \cdot 240) = 3.14155793\, rad \tag{2}$$

$$fov_h = 2 \cdot \arctan(\frac{W}{2} \cdot f) = 2 \cdot \arctan(\frac{640}{2} \cdot 240) = 3.14156661\, rad \tag{3}$$

The resulting image can be seen in Figure 4

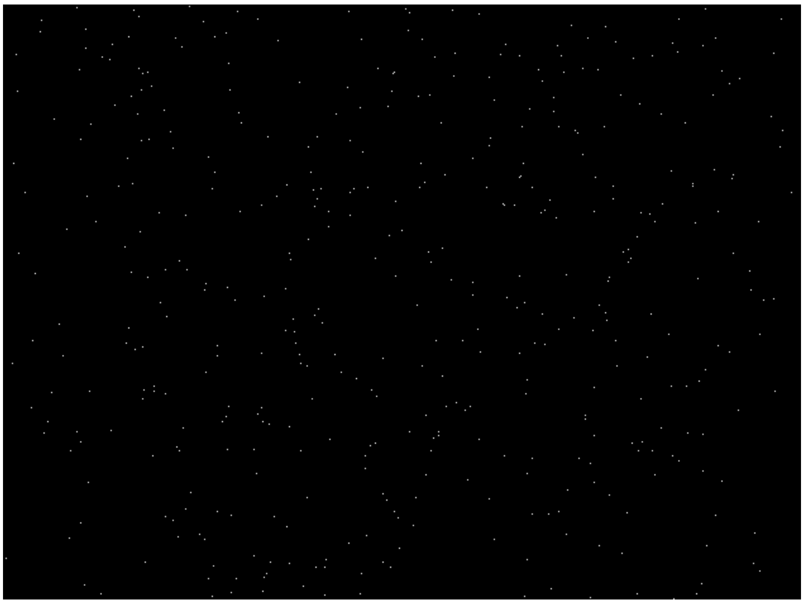

Figure 4: Image of stars in a 3D grid projected onto a 2D screen. All stars with same brightness.

3.2 Motion

After the still image was created, the next step was to animate the stars. Here two issues needed to be solved. The animation itself for each of the stars and the fade effect of each star when approaching the camera. The animation was solved by moving the stars towards the camera in the z-direction, adding +1 for each timestep in the update function. We also added a reset so that the stars were teleported back if they passed the camera so that the animation could be looped in a smooth manner.

Figure 5: Still image of the animated stars with fade of stars implemented. Stars with varying brightness.

The fade effect was implemented easily by just adding the code snippet specified in the assignment, see Figure 5.

Workload Distribution

The lab was done on one computer of the KTH computer rooms when all three members were present. We took turns in being the driver while the two others were navigators. We discussed the problems and then decided together on how to write the code. This report was written after the lab was completed, with references to the code and progress images, where its content was discussed prior to being written.