DH2323
Computer Graphics and Interaction
Lab 1: Introduction to 2D and 3D Graphics

Viktor Meyer viktorme@kth.se

June 15, 2020

1 Setup of the Lab Environment

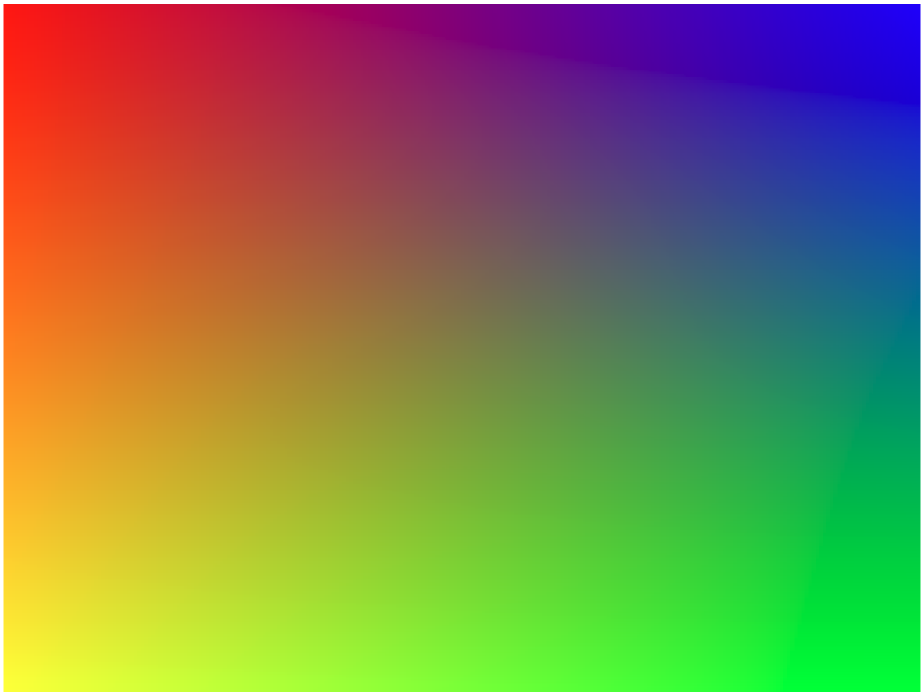
A personal computer was used for this lab which meant that all of the external libraries had to be setup. These two libraries were SDL and GLM. There were some problems with installing SDL since there are two versions that are not compatible with each other. Initially SDL2 was installed which led to some confusion as to why things did not work, although this was later resolved by installing SDL1.2 instead. GLM was already included in the CMake project and no problems were encountered. All things considered the setup was easy and straight forward.

2 Introduction to 2D Computer Graphics

Linear interpolation is a central concept in computer graphics programming. Given a number A, a number B and a timestep t, a linear interpolation gives a value from A to B weighted by t. Using $t = 0$ the interpolation value is A, using $t = 0.5$ the interpolation value is $(B - A)/2$ and using $t = 1$ the interpolation value is B. The lab required an implementation of linear interpolation that could create n discrete steps between A and B. This was straight forward since we can simply evaluate the standard linear interpolation function at incremental timesteps of $1/n$ to solve this problem.

In addition to implementing linear interpolation for scalar values, a task was to also support vector types. This was trivial since linear interpolation for vectors is as simple as interpolating component-wise.

With linear interpolation for both scalar and vector types supported, the next task was to interpolate colors across a surface:



To render the above image, the left side is interpolated from red to yellow and the right side is interpolated from blue to green. Once these two sides are known, the final step is to for each row interpolate between the sides to generate the intermediate pixels.

3 Star field

The star field effect is a classic effect that is perhaps most notably known from Star Wars. The effect simulates stars moving towards the camera along the z-axis.

To start, it is reasonable to initialize a set of stars (1000) to random starting positions. Once this is done, movement is simple since to simulate one could subtract the z-component of each star by $v * dt$ each tick. Where dt is the deltatime, that is the time since last tick.

Finally, to render the moving stars we need to transform the three dimensional coordinates to two dimensional screen space. This is known as projection and can be achieved by implementing a simple pinhole camera. Each coordinate of the stars are projected to two dimensional space and rendered using PutPixelSDL. For added effect, the color of each star changes depending on the distance from the camera, such that closer stars appear brighter than stars that are far away.