

# Project: OpenGL Version of Lab 3

Computer Graphics with Interaction DH2323

Author: Niklas Blomqvist, nblomqvi@kth.se

Lab partner: Annika Strålfors

November 12, 2014

## Contents

<b>1 Introduction</b>	<b>1</b>
1.1 Project specification . . . . .	1
<b>2 Method</b>	<b>1</b>
2.1 Part 1 . . . . .	1
2.2 Part 2 . . . . .	1
2.3 Part 3 . . . . .	2
2.4 Part 4 . . . . .	2
<b>3 Reflections</b>	<b>2</b>
<b>4 References</b>	<b>4</b>

# 1 Introduction

We weren't satisfied with the slow performance of rendering the room in lab 3 so for our project we wanted to give OpenGL a try. Our previous experience with OpenGL is non-existent and we are both new to the field of computer graphics. We have a background in computer science and programming.

## 1.1 Project specification

The goal of the project was to re-create lab 3 in OpenGL. OpenGL utilizes the graphics processing unit (GPU) rather than the central processing unit (CPU) for hardware-accelerated rendering which thus increases performance. Due to our lack of experience with OpenGL we decided to redo lab 3 step-by-step by following the lab instructions but making the necessary implementations in OpenGL instead.

We used OpenGL 3.0 and SDL 1.2 for this project and the reworked files from lab 3 are "CMakeLists.txt", "SDLauxiliary.h" and "skeleton.cpp". The primary code is written in C++. A small video presentation ("demo.avi") of the project was created and added to the submitting to demonstrate how the program works.

# 2 Method

The work was divided between us and we spent quite some time doing research. When we encountered problems we helped each other and tried to solve them together.

## 2.1 Part 1

The first part of the project consisted of researching OpenGL and following a few tutorials to get a basic understanding on how OpenGL works. We did also need to research how to set up the environment and compile everything. This was done by reworking the "CmakeLists.txt"-file to include OpenGL.

## 2.2 Part 2

The second step was to set up the room in OpenGL. Most of the initiations are found in the "SDLauxiliary.h"-file but not all. The lighting for example, needed to be initiated after the "gluLookAt"-method, which is called in the draw-function, in order for the lighting to be stationary and not moving as the observer(camera) moves around in the room.

### 2.3 Part 3

The third step was to actually render the room. This is done in the draw-function where we first clear the screen and buffer and load the identity matrix. We set the camera angle with "gluLookAt" and enable the lighting. The rendering is done between the glBegin(GL\_TRIANGLES) and glEnd()-functions where we iterate the triangles-list and color each pixel, similar to how it's done in lab 3. We end the draw-function by updating the screen.

### 2.4 Part 4

In the fourth and last step of the process we reworked the update-function. This consisted mainly of modelling the camera movement and a vector in the camera's direction was created to do this. The observer can walk around in the room and look up or down. Some unrealistic movement was added so the observer can move up and down to be able to view the scene from all angles. Lastly a light switch-function was added so the user can toggle the lighting.

The code is well-documented and is found with the submitted report.

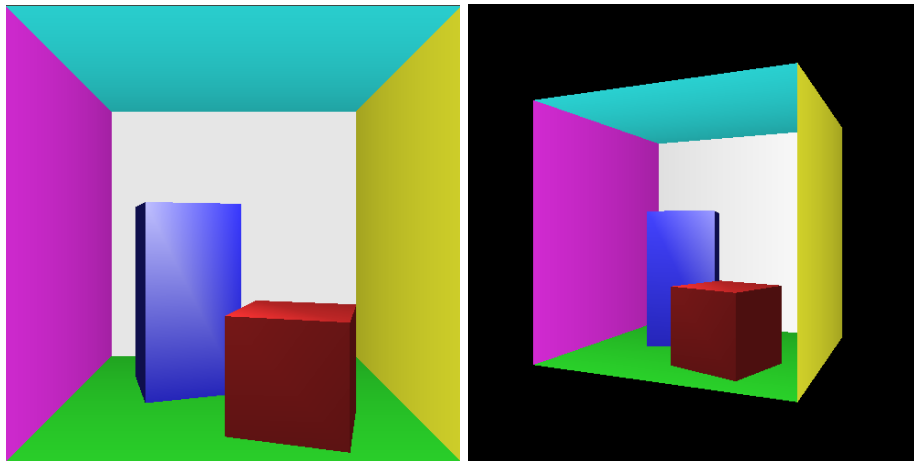


Figure 1: The outcome of the project.

## 3 Reflections

This project was very time-consuming and most of the time was spent doing research on the different tools used and possible features. The hardest part was to get everything working; setting up the environment with SDL and OpenGL and initiating the first functions and variables. The learning curve was big and

it took time to get somewhat confident to do the actual coding.

We wanted to implement some more features but felt that we didn't have the time to do it. What we especially wanted to do was to add textures to improve the graphical parts so that the room would look better than in lab 3. We had some other ideas on implementing sound and camera movement with the mouse and we wanted to improve the lighting in the room to look more realistic. We did some research and tried to implement better lighting, add sound and textures but had a hard time doing it.

The performance of the movement in the room was increased which can be seen in the demonstration video. OpenGL has built-in functions that help create the graphics we had to manually code in lab 3. An example is the the depth buffer which was pretty much one line, OpenGL did the rest.

I have learnt a lot from this project. I have gotten more insight in OpenGL, Cmake, Make and C++. Possible improvements could be, as mentioned above, improved lighting, added sound or textures and implementing mouse-movement. What I would have done differently if I had the insight I have now is probably to learn the basics of CMake and OpenGL more thoroughly from the start instead of going ahead thinking quickfixes would do it. I would probably have spent more time doing tutorials in OpenGL to get more familiar with it from the start.

As I mention above, the performance is increased enormously by using OpenGL compared to lab 3. Everything runs much smoother in this implementation and the rendering is a lot faster. The demo of the program can be found in the same folder as this report with the name "demo.avi".

## 4 References

**SDL and OpenGL tutorials:**

<http://lazyfoo.net/>

<http://www.libsdl.org/release/SDL-1.2.15/docs/html/guidevideopengl.html>

<http://ogldev.atSPACE.co.uk/>

**OpenGL documentation:**

<https://www.opengl.org/documentation/>

**C++ tutorials:**

<http://xoax.net/cpp/>