Virtual Reality

The Engine and The Robot Arm

**Summary**

[1 Introduction 3](#_Toc102552633)

[1.1 Abstract 3](#_Toc102552634)

[2 Arrangement 3](#_Toc102552635)

[2.1 Development Approach 3](#_Toc102552636)

[2.2 Management of the workload 3](#_Toc102552637)

[3 Implementation 4](#_Toc102552638)

[3.1 The Engine 4](#_Toc102552639)

[*3.1.1* *Basic Structure* 4](#_Toc102552640)

[*3.1.2* *Shaders* 4](#_Toc102552641)

[*3.1.3* *Lighting* 4](#_Toc102552642)

[*3.1.4* *Program* 5](#_Toc102552643)

[*3.1.5* *Skybox* 5](#_Toc102552644)

[*3.1.6* *Stereoscopic Rendering* 5](#_Toc102552645)

[*3.1.7* *Sphere/Box/Frustum Collision* 5](#_Toc102552646)

[4 Problems 6](#_Toc102552647)

[4.1 The Engine 6](#_Toc102552648)

[*4.1.1* *Lights* 6](#_Toc102552649)

[5 Examples 7](#_Toc102552650)

[6 Sitography 8](#_Toc102552651)

# Introduction

## Abstract

*The goal of this project is the implementation of a piece of software that uses various techniques learned during SUPSI Virtual Reality course. Specifically, the project consists in the development of a graphic engine that supports stereoscopic rendering interfaced with SteamVR and later used in form of a library by an associated client. This engine is therefore able to render models via a file scene uploading and, overall illumination, is managed via per pixel lighting illumination model based on Blinn-Phong. The client needs to show and operate a 3D robotic arm both on a screen or in a virtual reality headset, with which the user can interact in real-time by moving its articulations and also being able to cling a ball. The engine is developed as a deepening of a pre-existing project, but it uses a much more modern rendering pipeline since OpenGL version, which this project makes vast use, deeply changes way of thinking computer graphics.*

# Arrangement

## Development Approach

Since a first version of this graphic engine already existed, we focused on first resolve the little problems it had and then moved on the new features implementation. The engine we chose to use was the “Dominelli, Napoles and Mauro” one because it was the less problematic and the most complete. We made vastly use of the agile work methodology, introducing stand-up meetings every week right after professor course theory part explanations, and our Backlog consisted in weekly To-Do’s directly gave by our teacher.

## Management of the workload

As wrote in the previous chapters, we had to firstly correct some pre-existing errors and lacuna of implementation. Everyone participated in this phase and, in less than a couple of weeks, everything was ready for the big develop sprint. Then, each participant worked in shifts on the new requisite entry established by professor’s weekly backlog. Unfortunately, since our university program is full and crammed, not everyone could collaborate on every functionality for this project, but we ensured that each member was updated with the main code changes. Developing feature after feature we obtained an overall complete and working product.

# Implementation

## The Engine

In this section, we will discuss the implementation of the software and analyze some of the peculiar choices we made throughout the development phase. We will start with a brief description of the basic structure and then move on to the classes that we believe are useful and powerful for a virtual reality graphic engine.

### *Basic Structure*

Paragraph

### *Shaders*

Paragraph

#### Vertex Shader

Paragraph

#### Fragment Shader

Paragraph

### *Lighting*

Paragraph

### *Program*

Paragraph

#### VAOs & VBOs

Paragraph

### *Skybox*

Paragraph

### *Stereoscopic Rendering*

Paragraph

#### FBOs

Paragraph

#### OpenVR

Paragraph

### *Sphere/Box/Frustum Collision*

Paragraph

# Problems

## The Engine

### *Lights*

#### Multiple lights rendering

Paragraph

#### *Skybox*

Paragraph

# Examples

In this chapter there are some screenshots of the application while running.

Here below, the view shows different angles, and with the arm and the ball in different positions.

# Sitography

* + - 1. *GLM documentation:* <https://glm.g-truc.net/0.9.2/api/index.html>
      2. *C++ reference manual:* <https://en.cppreference.com/w/>
      3. *OpenGL documentation manual:*

<https://www.khronos.org/registry/OpenGL-Refpages/gl4/>