Interactive Graphics HW1

Andrea Napoletani 1609711

April 2020

1 Introduction

In this report will be described the development of the HW1's implementation. There will be described the main features and the most important technical choices taken for the project.

2 Geometry

The initial cube has been replaced with an irregular geometry composed of 25 vertices and 18 faces. Each face of the geometry is a polygon of 4 vertices declared in the application file. In order to render this object with WebGL primitives, each face is divided and represented as two triangles. To all the vertices of the triangles that composed a face is assigned the same normal vector calculated from one of them (example by vertices a, b and c) as

$$n = (c - a) \times (b - a)$$

The order of the vertices given as parameters to the function describes the orientation of the face according to the right-hand rule.
Each vertex of a face also have an assigned texture coordinate (declared in an array inside the application) in order to apply an image as a texture (details in the next paragraphs). All of this attributes of the vertices will be passed from the application to the vertex shader in the HTML file in order to set the position of the vertex and calculate vectors for the application of the light effects.

3 Viewer position and projection

In this section some sliders have been added to allow the user to tune parameters related to the view volume and the viewer position. These parameters are related to the computation of the modelViewMatrix through the ***lookAt(eye, at, up)*** function (in particular the attributes *radius*, *theta* and *phi* that composed the parameter *eye*) and the projectMatrix (perspective projection) through the parameters *fovy*, *aspect*, *near* and *far*. Both these matrix have been calculated inside the application and then passed to the HTML instead of compute it directly in the vertex shader, in this way they are computed just one time instead of compute them for each vertex.

4 Lights and Material

In the scene there are two different light sources, one is directional and one is a spotlight. Both have two components, *ambient* and *diffuse* (*specular* component is omitted according to our custom cartoon shading model). Some parameters of the spolight can be tuned by the user exploiting the sliders at the upper part of the page. In particular the user can change the *direction* of the light (tuning the components x, y and z), the *cutOff* value which allow to decide the angle of the light cone (so the ”size” of the cone) and the *alpha* component that is related to the *attenuation* factor of the spotlight. Materials parameter have been assigned to the object: *materialAmbient* and *materialDiffuse*.

5 Shading

A per-fragment shading model has been applied to the object. C_i and C_s factors have been computed according to the Simple Cartoon Shading model given in the assignment. This factors have been computed for both the light sources and have been used as color of each fragment (different cases according to the presence or not of the spotlight's effect).

6 Texture

An image of bricks has been added as a texture to the object. It is applied to each fragment multiplied for the coefficent(s) of the light(s) in a way to obtain a combination of both effects. Texture can be disabled clicking on the button at the top of the page.