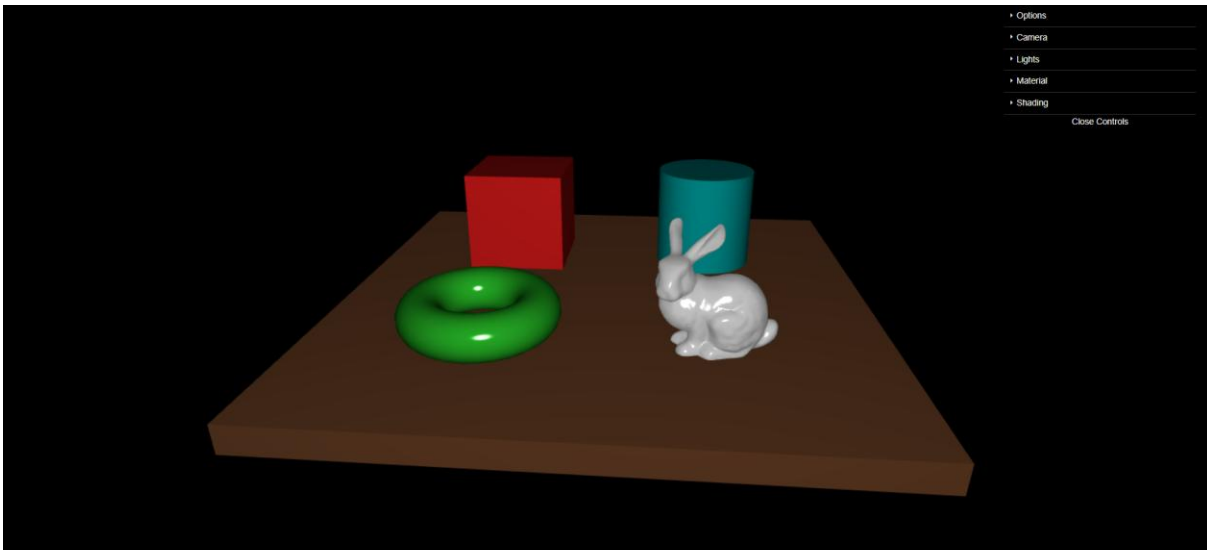
Project 3 CGI



Francisco Rodrigues 67753

p.2

Shader Variables

Phong:

Vertex shader

```
const int MAX_LIGHTS = 8; // max number of lights
in vec4 a_position; // vertex position in modelling coordinates
in vec4 a_normal; // vertex normal in modelling coordinates
uniform mat4 u_mModelView; // model-view transformation
uniform mat4 u_mNormals; // model-view transformation for normals
uniform mat4 u_mProjection; // projection matrix
out vec3 v_normal; // normal vector in camera space
out vec3 v_viewer; // View vector in camera space
```

Fragment shader

```
const int MAX_LIGHTS = 8; // max number of lights
struct LightInfo {
   // Light colour intensities
   vec3 ambient;
   vec3 diffuse;
   vec3 specular;
   float brightness; // Light brightness multiplier
   vec4 position;  // Position/direction of light (in camera coordinates)
   // Spotlight properties
   bool is_spotlight; // Is this light a spotlight
   vec3 axis;      // Direction of spotlight axis
   float aperture; // Spotlight aperture
   float cutoff;   // Spotlight cutoff
};
```

```
// Material properties

struct MaterialInfo {

   vec3 Ka; // ambient factor

   vec3 Kd; // diffuse factor

   vec3 Ks; // specular factor

   float shininess; //shininess of the material

};

uniform int u_n_lights; // Effective number of lights used

uniform LightInfo u_lights[MAX_LIGHTS]; // The array of lights present in the scene

uniform MaterialInfo u_material; // The material of the object being drawn

in vec3 v_normal; // Normal vector at the surface point

in vec3 v_viewer;  // Vector from surface point to viewer

out vec4 color; //color of the pixel
```

Gouraud

Vertex shader

```
const int MAX_LIGHTS = 8; // max number of lights

struct LightInfo {

   // Light colour intensities

   vec3 ambient;

   vec3 diffuse;

   vec3 specular;

   float brightness; // Light brightness multiplier

   vec4 position;  // Position/direction of light (in camera coordinates)

   // Spotlight properties

   bool is_spotlight; // Is this light a spotlight

   vec3 axis;      // Direction of spotlight axis

   float aperture; // Spotlight aperture

   float cutoff;   // Spotlight cutoff

};

// Material properties

struct MaterialInfo {

   vec3 Ka; // ambient factor

   vec3 Kd; // diffuse factor

   vec3 Ks; // specular factor

   float shininess; //shininess of the material

};

uniform int u_n_lights; // Effective number of lights used

uniform LightInfo u_lights[MAX_LIGHTS]; // The array of lights present in the scene

uniform MaterialInfo u_material;       // The material of the object being drawn

in vec4 a_position; // vertex position in modelling coordinates

in vec4 a_normal; // vertex normal in modelling coordinates
```

uniform mat4 u_mModelView; // model-view transformation

uniform mat4 u_mNormals; // model-view transformation for normals

uniform mat4 u_mProjection; // projection matrix

out vec3 v_color; // color to be passed to the fragment shader

Fragment shader

in vec3 v_color; // color calculated

out vec4 color; // final color

Extra implementations

Spotlight on objects

By selecting 1, 2, 3 or 4 on the keyboard, a spotlight appears on the selected object, changing the values of the first light in the gui.

1 – Cube

2 – Cylinder

3 – Torus

4 – Bunny

Lightshow

When pressed “p”, a short “lightshow” starts, showing a variation of spotlights with different colors on different objects.

Scenegraph JSON

The scenegraph is represented in a tree in a json file, each node has:

- Name
- Type
- Transform
    - Translation
    - Rotation
    - Scale

Where the type is chosen between internal and leaf, and the transformation to apply are specified in the field in it. Leaf nodes have also a primitive, which indicates the primitive to use to draw it, and the material type, for each type (ambient, diffuse, specular) and the shininess. Internal nodes have children, and those will have the same translations already applied by the parents. The world has also an additional field with the lightprimitive, which stores a light leaf with the usual variables and an additional one, the currentIndex, which is used when drawing a light, to know who the last parent was. This leaf node is used to add lights to the scene, which will result in adding an internal light node with this leaf as a child.