Lesson 6 :
Light & Material

Introduction

Light is a thing that make scene real. Effectively, in a real world, light is anywhere present.
Objects have also their own reaction in presence of a light.

This lesson show how OpenGl use this two things : Light & Material
Tutorial 12 to 15 are in relation to this lesson.

Notion of Light

OpenGL approximate the light as 4 independant components.
When light is enable, the color of a vertex depends on his color and also on the 4 following color components of the light :

- Diffuse (GL_DIFFUSE) : light ray comes from a light source and hit directly a surface.
  The diffuse color of a vertex depends on his position (like all the component's colors) and his orientation in relation to the light source.

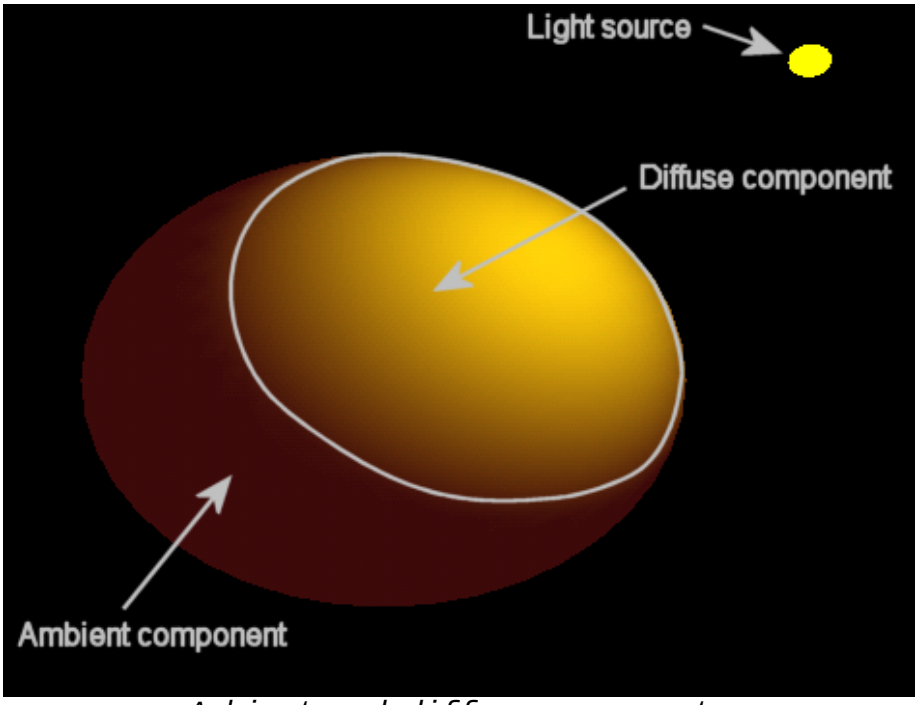
- Ambient (GL_AMBIENT) : all the vertex of the scene are affected by this component.
  When a light ray hit a surface, it is scattered equaly in all directions. After a lot of bounce onto surfaces, the light seemed to come from anywhere.
  The ambient component correspond to this physical phenomenon.

- Specular (GL_SPECULAR) : specular is the light component that create highlight on surfaces like mirror and metal.
  This component should be used only with material, see Tutorial 14 for this.

- Emission (GL_EMISSION) : some objects like lamp, LED ... can emits light. Emission is used to creates this kind of objects.

We will see on the light calculation part how OpenGl calculates the vertex color with these light components.

This picture show you the effect of the ambient and diffuse light components :



Ambient and diffuse components

Light source and material properties

Light source is often associated with object that have material properties.

A light source emits light in **all** the directions. It is composed of a diffuse, ambient and specular component.
Each light source is associated with the OpenGl constant :
    GL_LIGHT*i*
i is an integer number between 0 to 7, that is to say we can use 8 differents light sources.
Light source should be view in Tutorial 12-13.

Material properties defines how the object react with light, that is to say how the object absorb, reflect ... the light. Materials should be view in the Tutorial 14-15.

Using a light source

To defines the light source properties, we use the following method :
    gl.**glLight*(***light, propertie, value***)**
***** is the type of value (ex: i for int, iv for int[], f for float ...)
light is the constant GL_LIGHT*i* that refer to the light source
propertie is the propertie to set
value is the value to affect to the *propertie*

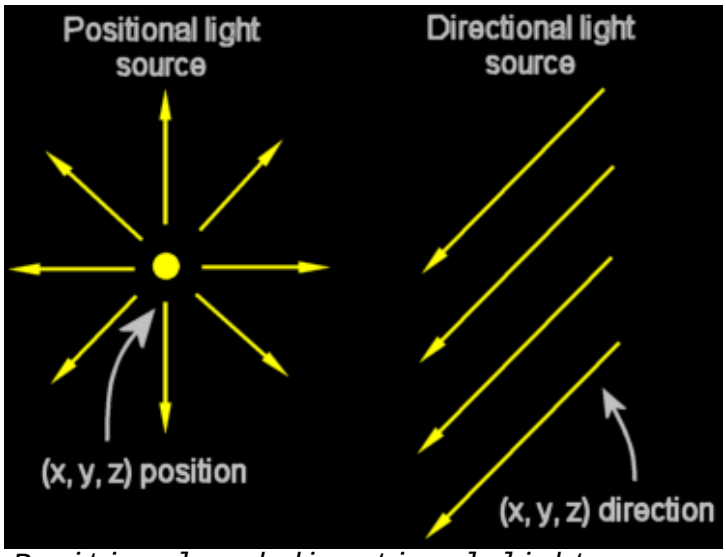
When we use light, we need to orientate the surface of our objects. See Tutorial 12 for more informations about this.

Position

The position of the light is define with :
    gl.**glLightfv(**GL_LIGHT*i*, GL_POSITION, *position***)**
position is a float array that contains the x, y, z and w position of the

light.

The light source can be a **positional** (w > 0) or **directional** (w = 0) light source depending on the w value.
A positional light source is positioned at the location (x, y, z). The source emits light from that particular location towards all directions. It is used to simulate lamp, bulb ...
A directional one haven't got any location. The source emits light from an infinite location, the ray are all parallel and have the direction (x, y, z). A directional is not subject to attenuation since it is at an infinite distance. It is used to simulate the sun for exemple.



Positional and directional light source

- *Ambient, diffuse and specular*

For each lights, we defines their ambient, diffuse and specular color. For a realistic light effect, the ambient and diffuse colors are generally the same.
    gl.**glLightfv(**GL_LIGHT*i*, GL_AMBIENT, *ambientColor***)**
    gl.**glLightfv(**GL_LIGHT*i*, GL_DIFFUSE, *diffuseColor***)**
    gl.**glLightfv(**GL_LIGHT*i*, GL_SPECULARE, *specularColor***)**
ambientColor, *diffuseColor* and *specular* are float array that contains the rgba value of the color.

- *Light attenuation*

In a real world, the light intensity should decrease in inverse proportion to the distance.

In OpenGl, we also can defines the light attenuation with the distance. OpenGl calculates an attenuation factor (between 0 and 1) that is multiplied to the ambient, diffuse and specular color. By default, they are no attenuation (attenuation factor is 1) so you have to defines your own attenuation.

The attenuation formula is (defult attenuation factor is 1) :
    attenuation factor = 1/(GL_CONSTANT_ATTENUATION + GL_LINEAR_ATTENUATION*d + GL_QUADRATIC_ATTENUATION*d^2)
d is the distance between the light and the vertex.

```
GL_CONSTANT_ATTENUATION  : default value is 1
GL_LINEAR_ATTENUATION    : default value is 0
GL_QUADRATIC_ATTENUATION : default value is 0
```

To have an attenuation with the distance, you should set to a value different to 0 the linear or quadratic attenuation.
Here is an exemple of attenuation with distance :

```
    gl.glLightf(GL_LIGHTi, GL_CONSTANT_ATTENUATION, 1.0f)
    gl.glLightf(GL_LIGHTi, GL_LINEAR_ATTENUATION, 0.2f)
    gl.glLightf(GL_LIGHTi, GL_QUADRATIC_ATTENUATION, 0.08f)
```

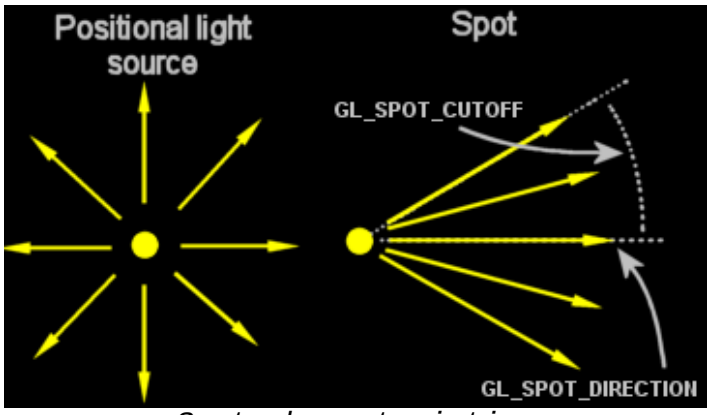
Particular light source

We have seen that positional source emits light in all the direction. But, we can creates a **spot** that emits lights into an emission cone by restricting the emission area of the light source.

To creates spot, we defines two properties in addition to those of a positional light source :

```
    GL_SPOT_DIRECTION : direction of the cone emission's axis
    GL_SPOT_CUTOFF    : maximum angle of a ray in relation to the spot direction
                        value in degree on the interval [0, 90]
```

These properties are represented in the following picture :



Spot characteristics

We can also defines the attenuation of the light depending of the angle of the ray between the center of the cone.
GL_SPOT_ATTENUATION defines how the light is concentrated in the center of the cone. 0 is equivalent to no attenuation. I think that the largest attenuation is 128.
Higher the value is, more the light is concentrated in the center of the shape (cone).

Lighting model

We can defines 3 things with the lighting model :

- GL_LIGHT_MODEL_AMBIENT : set the global ambient color
- GL_LIGHT_MODEL_LOCAL_VIEWER : use a local or infinite viewpoint
- GL_LIGHT_MODEL_TWO_SIDE : defines if we use a two sided light or a one sided light

Lighting model properties are setted with :

gl.**glLightModel*(***propertie, value***)**
propertie is the propertie to set
value is the value to affect to the *propertie*

Values

The value for GL_LIGHT_MODEL_AMBIENT is a color so an array that contains the rgba values.

GL_LIGHT_MODEL_LOCAL_VIEWER defines how specular component is calculated. The specular highlight depends on the direction from the vertex and the viewpoint and the direction from the vertex and the light source. So, the highlight depends on the eye position.

With an **infinite viewpoint**, the direction from the vertex and the viewpoint remains the same for all vertex so this direction don't need to be calculated. Due to this reason, the calculation of the light is faster. This is the default viewpoint.
With a **local viewpoint**, the direction between the viewpoint and the vertex need to be calculated. It creates a more accurate model for the calculation of the highlight.
0 = infinite viewpoint
1 = locale viewpoint

GL_LIGHT_MODEL_TWO_SIDE defines if the light is applyed one the two sides of an object or only on one side.
0 = one sided
1 = two sided

Using Material

Material is a notion that is associated with an object.
This notion defines how the object react with a light : object's absorbtion, reflection ...

For exemple, some object can absorb a particular color or reflects light (ex: mirror and metal).

Material properties

Materials properties are similar to light propertis. Here are the differents properties we can assign to an object :
- GL_AMBIENT : color of the ambient reflection
- GL_DIFFUSE : color of the diffuse reflection
- GL_SPECULAR & GL_SHININESS : color of the specular reflection. This reflection defines if the light produce an highlight on the object like in mirror and metal. Shininess is used for the dimension and intensity of the highlight.
- GL_EMISSION : give the effect that the object **seem** to emit light. This is used to creates lamp or bulb.
  WARNING : the object **seem** to emit light but emits **NO** light. You need to creates a light source in the position of the object to really emit lights (see Tutorial 15 for an application of this).

These properties defines the same thing than for light, look at the

beginning of this Lesson. Tutorial 14 shows the effect of all of these properties.

Setting a property

The method we have used for light is **glLight**, for **material** we use **glMaterial** :
gl.**glMaterial*(***face, propertie, value***)**
face is the face : GL_FRONT, GL_BACK or GL_FRONT_AND_BACK
propertie is the propertie to set
value is the value to affect to the *propertie*

Applying a material to an object

The material property affected to the object drawn is the current values of the differents properties. So, if you set one time the properties to a specific value, all the objects drawn after that should have this propertie until you change a value.

So, to apply a material to an object, you need to set the wanted material. After, draw the object.
Warning, if you have many objects that have differents material properties, I recommend you to set all material properties for each object to avoid to apply a property that is not wanted.

Light, Material and Mathematic

This part show you how OpenGl use all the properties view previously to calculate the color of an object.
You need to read this part to understand the relations between these properties.

Color of a vertex

Here is the global formula used to determine the color of a vertex :
color = GLOBAL_AMBIENT + AMBIENT + DIFFUSE + SPECULAR + EMISSION

Each term of this equation is caculated with the following expressions.

GLOBAL_AMBIENT Term
This term is the ambient component of the light model attenuated by the ambient property of the material :
GLOBAL_AMBIENT = GL_LIGHT_MODEL_AMBIENT * GL_AMBIENT from the material

AMBIENT DIFFUSE SPECULAR Term

Ambient term is the sum of the ambient lights that come from each lights and the global ambient from the light model.
AMBIENT + DIFFUSE + SPECULAR = attenuationFactor * spotFactor * (GL_AMBIENT from the material*GL_AMBIENT from the light + GL_DIFFUSE from the material*GL_DIFFUSE from the light*.... + GL_SPECULAR from the material*GL_SPECULAR from the light*GL_SHININESS*...)

```
spotFactor = 1 if the light is not a spot
             0 if the light is a spot and the vertex is outside the spot's
cone
             GL_SPOT_EXPONENT*...
```

Term represented by ... depends on the position of the vertex and its normal in relation with the light position.
You don't need to know exactly the formula, you only need to know that terms depends on the position of the vertex and its orientation (normal vector).

EMISSION Term
The emission term is the emission value of the material :
    EMISSION = GL_EMISSION

Previous Lesson Back Next Lesson