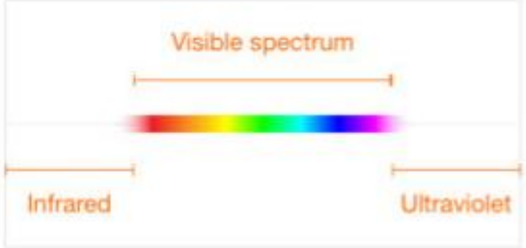
LIGHT IN CG

HAMZAH ASYRANI SULAIMAN

Light, electromagnetic radiation that can be detected by the human eye.

QUESTION #2

WHAT DO WE MEAN BY VISIBLE LIGHT?

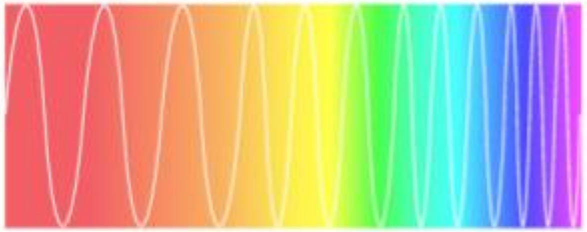


→ Visible light is the part of the larger spectrum of electromagnetic radiation that the naked human eye can detect. It is the light we normally talk about in everyday life.

Infographic: what is light

QUESTION #5

WHY DO DIFFERENT COLORS EXIST?



→ The light waves that we can see have different wavelengths. Red light has the longest wavelength, violet light has the shortest.

https://www.osram-lamps.com/get-inspired-by-the-fascination-of-lighting/infographic-what-is-

Light

- Light is form of energy that you can see.
- Light is on kind of radiant energy.
- Light comes from many different sources.
- Some items produce light while others reflect light.

Vision and Color Models

Light has three properties: **Wavelength, speed, and amplitude**.

The wavelength **determines** the type of **light** (color, etc.).

Speed is **determined** by whether **light** passes through a vacuum or some material. ... The more photons emitted per unit time, the greater the **intensity** of the **light**.

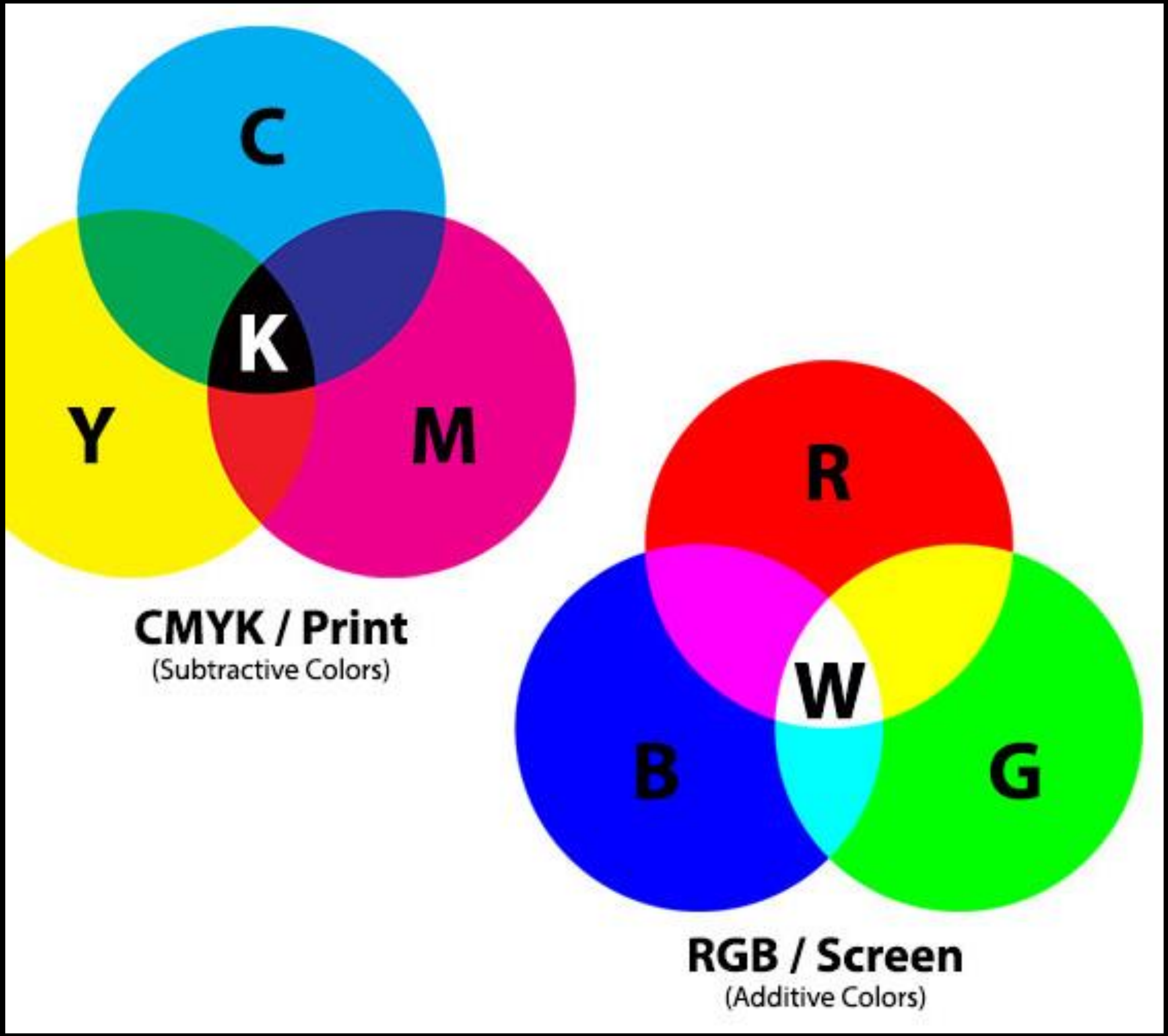
Vision and Color Models

The human eye can see **7,000,000 colors**. Some of these are eyesores. Certain colors and color relationships can be eye irritants, cause headaches, and wreak havoc with human vision. Other colors and color combinations are soothing.

RGB COLOR MODEL and COMPUTER GRAPHICS

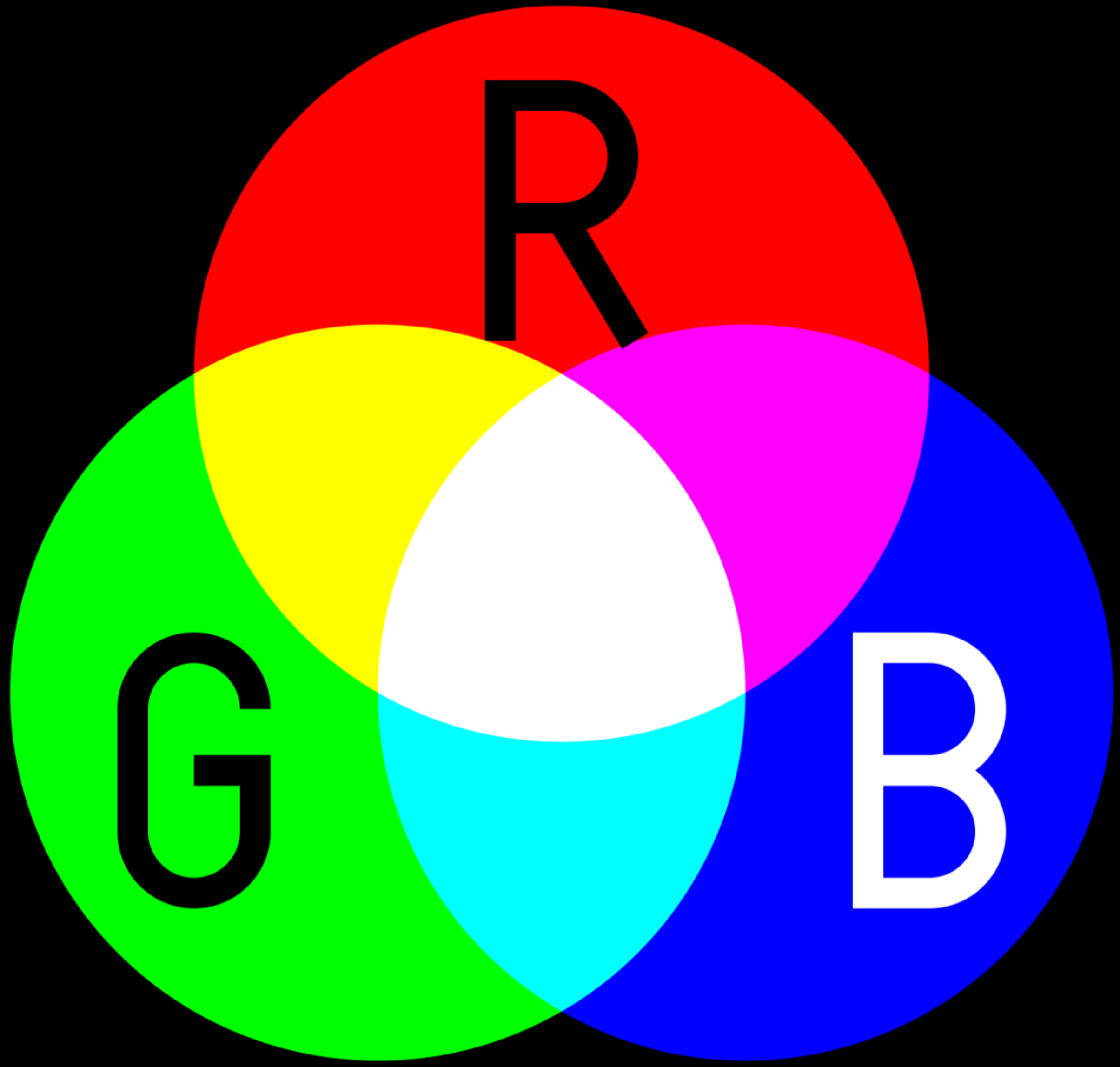
Colors perceived in subtractive models are the result of reflected light.

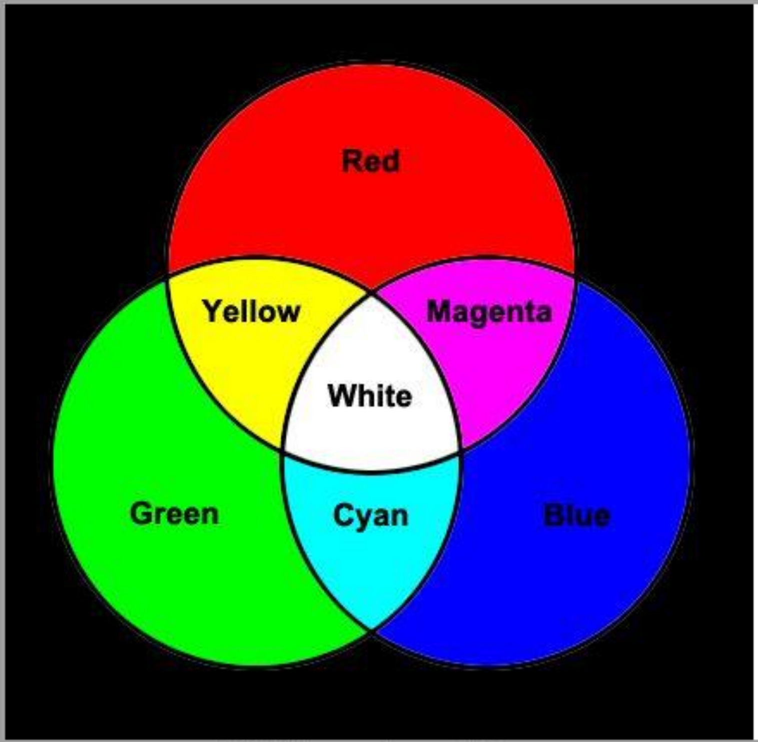
There are several established color models used in computer graphics, but the two most common are the **RGB** model (**Red-Green-Blue**) for computer display and the CMYK model (**Cyan-Magenta-Yellow-Black**) for printing



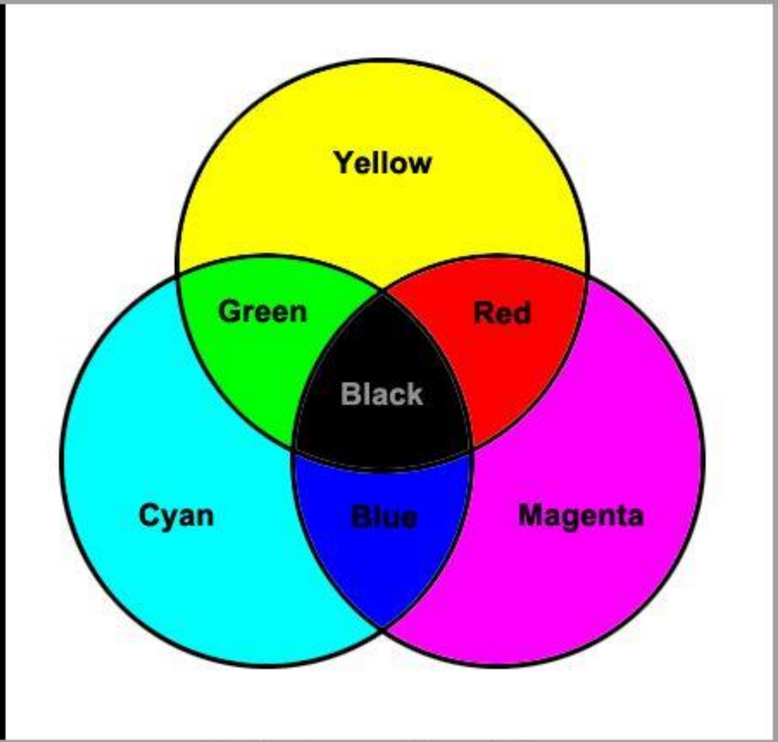
RGB COLOR MODEL

RGB (red, green, and blue) refers to a system for representing the **colors** to be used on a computer display. Red, green, and blue can be combined in various proportions to obtain any **color** in the visible spectrum. Levels of R, G, and B can each range from 0 to 100 percent of full intensity.





Additive color mixing

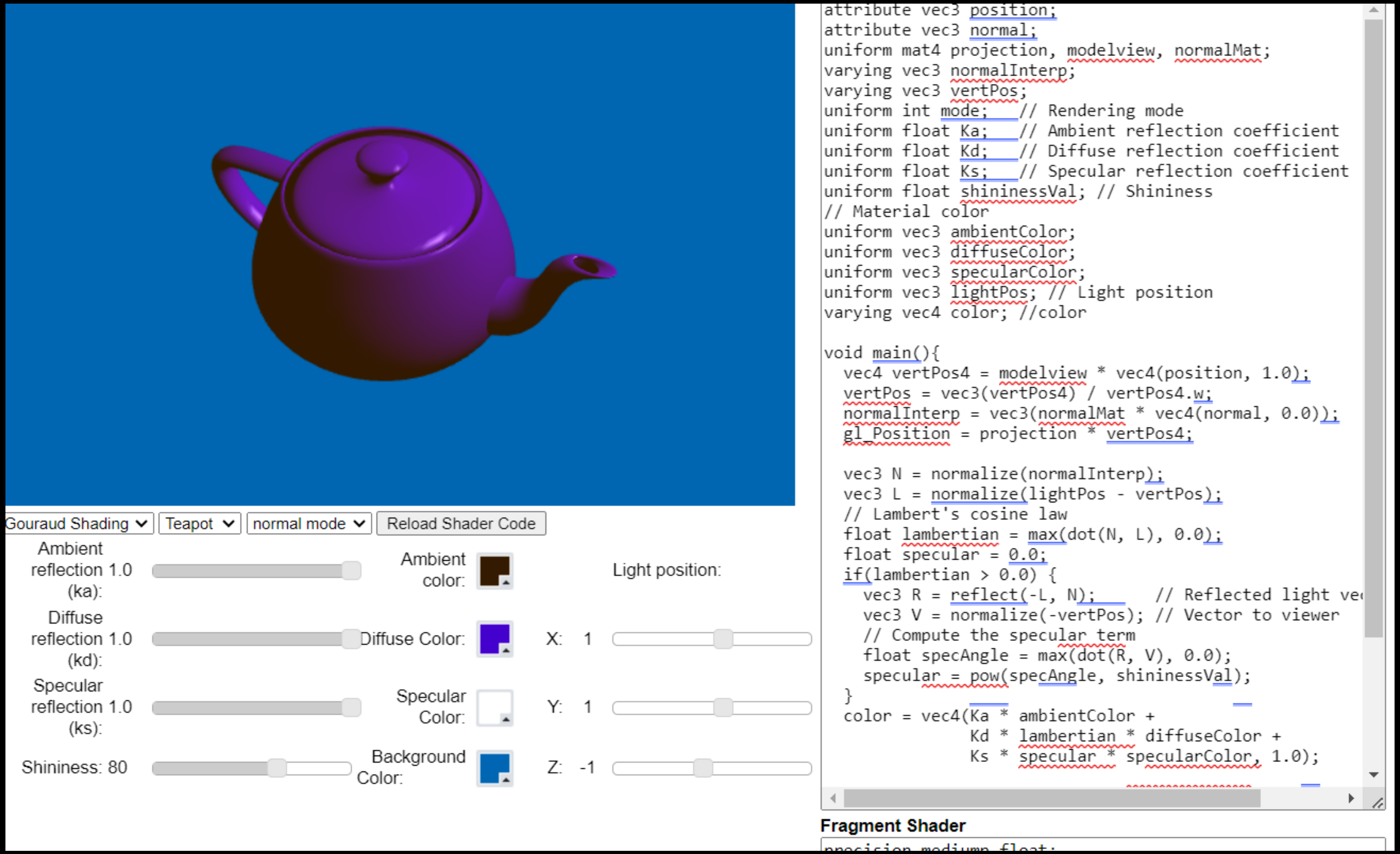


Subtractive color mixing

Additive color systems start without light (black). Light sources of various wavelengths combine to make a color.

Subtractive color systems start with light (white). Colored inks, paints, or filters between the viewer and the light source or reflective surface subtract wavelengths from the light, giving it color.

http://www.cs.toronto.edu/~jacobson/phong-demo/



LET'S CODE OPENGL LIGHTING

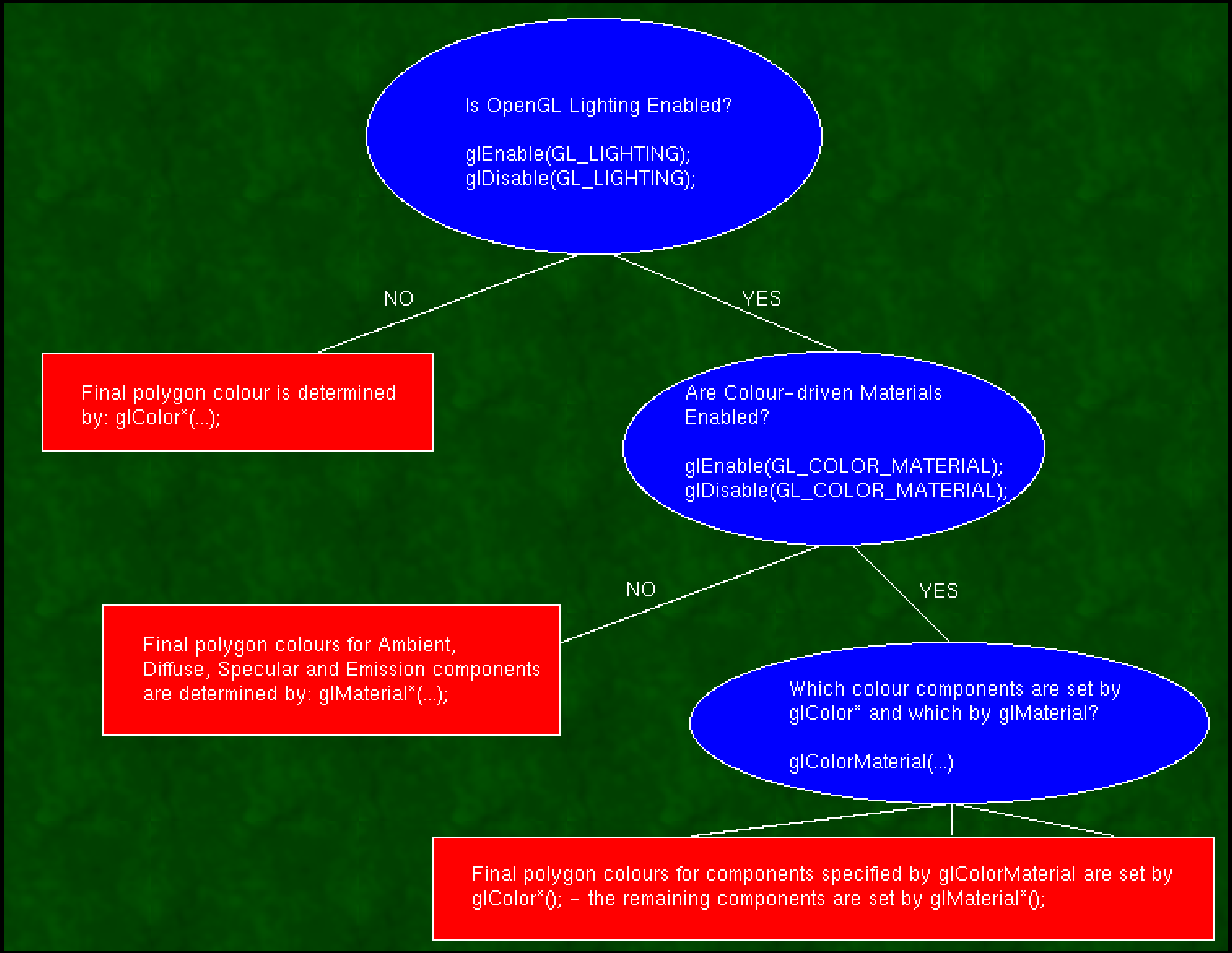
Lighting ENABLED or DISABLED?

The first - and most basic - decision is whether to enable lighting or not.

```
glEnable ( GL_LIGHTING ) ;
```

...or...

```
glDisable ( GL_LIGHTING ) ;
```


Is OpenGL Lighting Enabled?
glEnable(GL_LIGHTING);
glDisable(GL_LIGHTING);
NO
YES
Final polygon colour is determined by: glColor*(...);
Are Colour-driven Materials Enabled?
glEnable(GL_COLOR_MATERIAL);
glDisable(GL_COLOR_MATERIAL);
NO
YES
Final polygon colours for Ambient, Diffuse, Specular and Emission components are determined by: glMaterial*(...);
Which colour components are set by glColor* and which by glMaterial?
glColorMaterial(...)
Final polygon colours for components specified by glColorMaterial are set by glColor*(); - the remaining components are set by glMaterial*();

LET'S CODE LIGHTING

If it's **disabled** then all polygons, lines and points will be coloured according to the setting of the various forms of the glColor command. Those colours will be carried forward without any change other than is imparted by texture or fog if those are also enabled. Hence:

```
glColor3f ( 1.0f, 0.0f, 0.0f ) ;
```

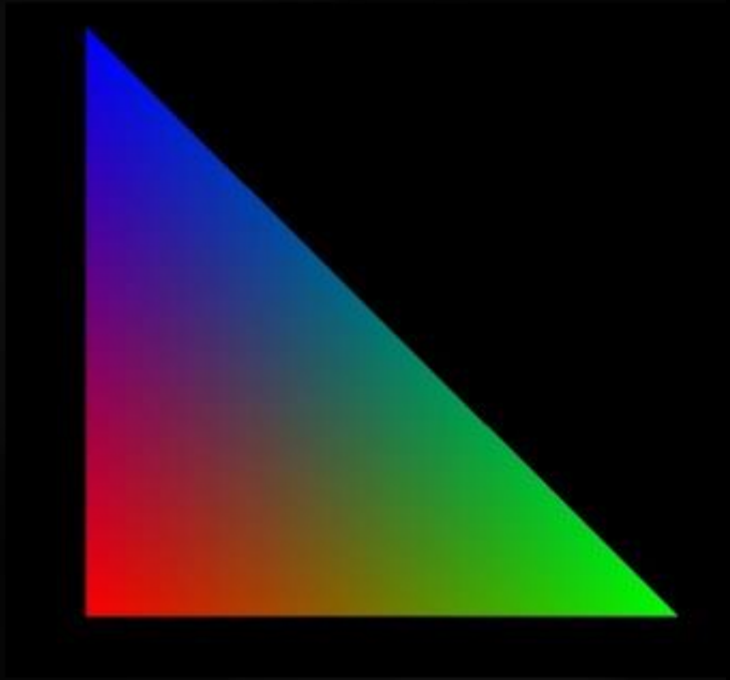
```
void GLWidget::paintGL()
{
    glClear ( GL_COLOR_BUFFER_BIT );

    // Reset the drawing perspective
    glLoadIdentity();

    // triangle
    glBegin (GL_TRIANGLES);
        GlColor3f ( 1.0,  0.0,  0.0 );
        glVertex3f ( 5.0,  5.0, 0.0 );

        glColor3f ( 0.0,  1.0, 0.0 );
        glVertex3f ( 25.0, 5.0, 0.0 );

        glColor3f ( 0.0,  0.0, 1.0 );
        GlVertex3f ( 5.0,  25.0, 0.0 );
    glEnd();
}
```



...gets you a pure red triangle no matter how it is positioned relative to the light source(s).

With GL_LIGHTING enabled, we need to specify more about the surface than just it's colour - we also need to know how shiny it is, whether it glows in the dark and whether it scatters light uniformly or in a more directional manner.

The idea is that **OpenGL switches over to using the current settings of the current 'material' instead of the simplistic idea of a polygon 'colour'** that is sufficient when lighting is disabled. We shall soon see that this is an over-simplistic explanation - but keep it firmly in mind.

Different Types of Light

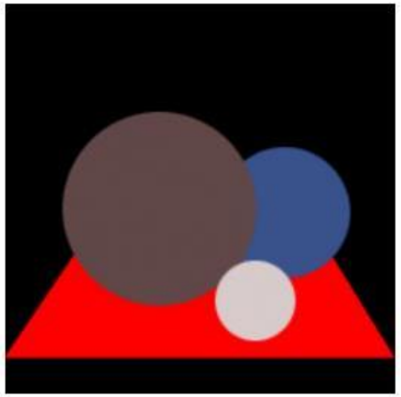
Specular: Sets the color for highlights

Diffuse: is the color of the object when it is illuminated

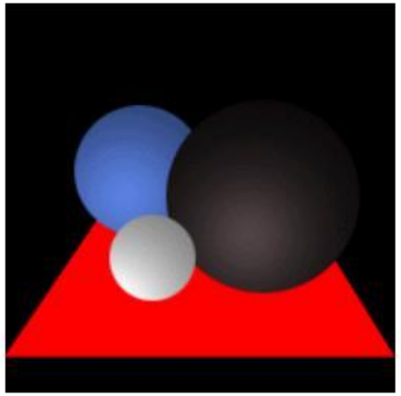
Ambient: is the color of the mesh when it's not illuminated

Emissive: is the type of light which is being emitted by an object

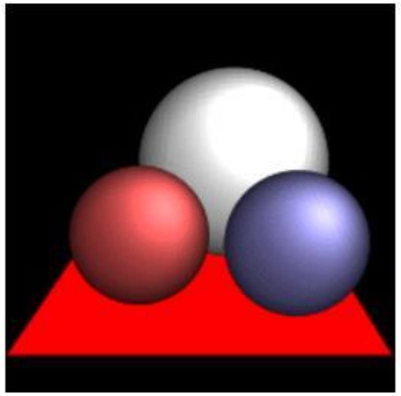
Types of Light and effects



1. Ambient light in a scene with 3 spheres.



2. Diffuse light hitting the surface of 3 spheres. Notice, the spheres look matte and almost plastic like.

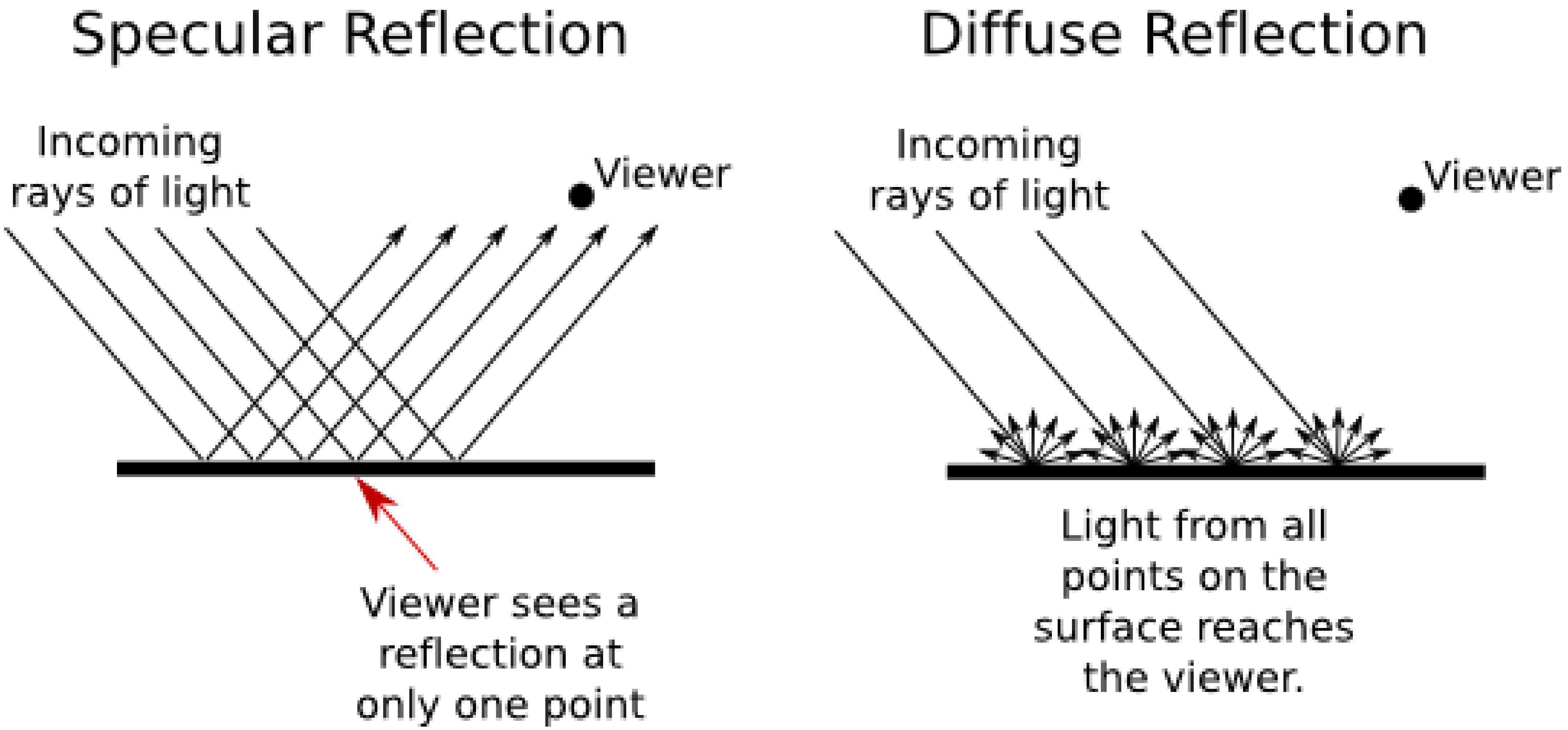


3. The three spheres illuminated by specular light. Imagine an extremely shiny billiard ball and the sheen it creates

The OpenGL light model presumes that the light that reaches your eye from the polygon surface arrives by four different mechanisms:

glMaterial and **glLight**

- **AMBIENT** - light that comes from all directions equally and is scattered in all directions equally by the polygons in your scene. This isn't quite true of the real world - but it's a good first approximation for light that comes pretty much uniformly from the sky and arrives onto a surface by bouncing off so many other surfaces that it might as well be uniform.
- **DIFFUSE** - light that comes from a particular point source (like the Sun) and hits surfaces with an intensity that depends on whether they face towards the light or away from it. However, once the light radiates from the surface, it does so equally in all directions. It is diffuse lighting that best defines the shape of 3D objects.
- **SPECULAR** - as with diffuse lighting, the light comes from a point souce, but with specular lighting, it is reflected more in the manner of a mirror where most of the light bounces off in a particular direction defined by the surface shape. Specular lighting is what produces the shiney highlights and helps us to distinguish between flat, dull surfaces such as plaster and shiney surfaces like polished plastics and metals.
- ***EMISSION** - in this case, the light is emitted by the polygon - equally in all directions.*

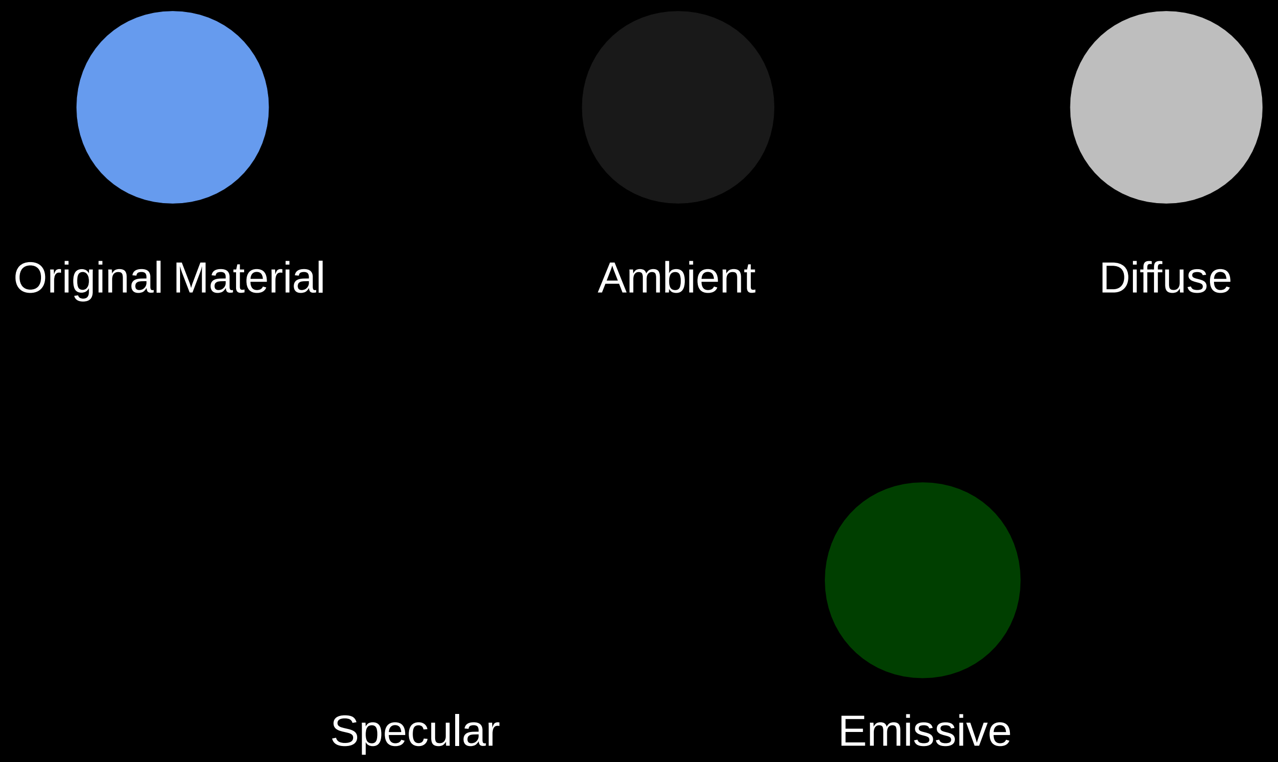

Specular Reflection
Incoming
rays of light
Viewer
Viewer sees a
reflection at
only one point
Diffuse Reflection
Incoming
rays of light
Viewer
Light from all
points on the
surface reaches
the viewer.

LET'S CODE LIGHTING

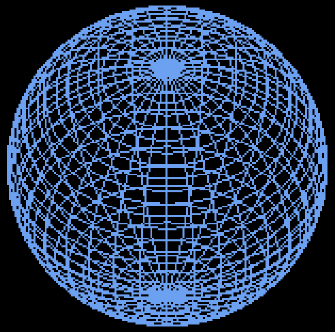
So, there are three common light colours for each light - Ambient, Diffuse and Specular (set with glLight) and four for each surface (set with glMaterial). Emissive is only mechanism not actual light.

All OpenGL implementations support at least eight light sources - and the glMaterial can be changed at will for each polygon (although there are typically large time penalties for doing that - so we'd like to minimise the number of changes).

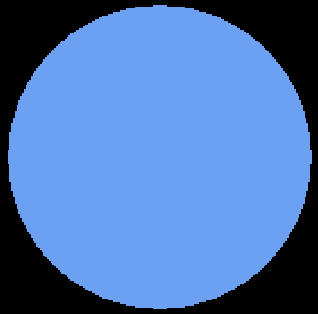
Color Used in Example



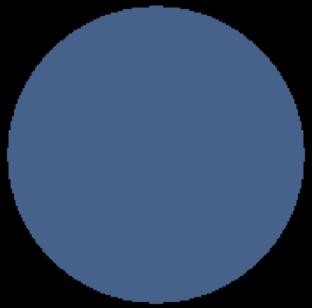
Example



Wireframe



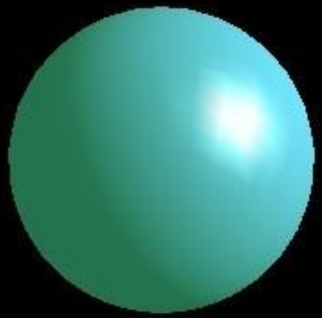
1. Original Color



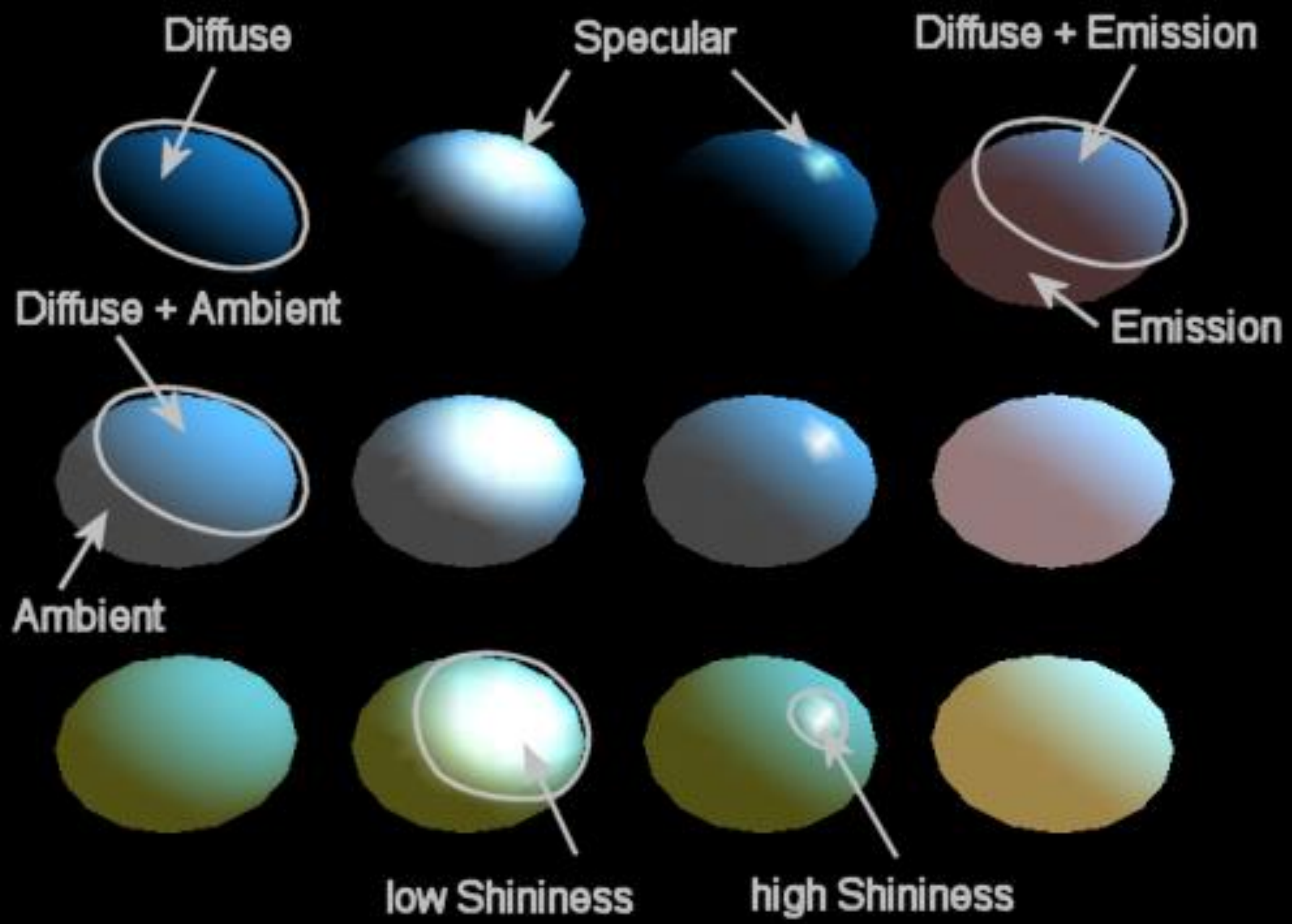
2. Original Color + Ambient

3. Original Color + Ambient + Diffuse

4. Original Color + Ambient + Diffuse + Specular



5. Original Color + Ambient + Diffuse + Specular + Emissive


Diffuse
Specular
Diffuse + Emission
Diffuse + Ambient
Emission
Ambient
low Shininess
high Shininess

- To use lighting in OpenGL we first must enable lighting with the command **glEnable(GL_LIGHTING)**
- In OpenGL we can define up to 8 light sources.
- We can enable each light source seperatly by using the command **glEnable(GL_LIGHT<0..7>)** .
- We can disable with the command **glDisable.**

LET'S CODE MATERIAL LIGHTING

From this point performing a glColor command (or setting the glColor via a vertex array or something) has the exact same effect as calling:

```
glMaterial ( GL_FRONT_AND_BACK, GL_EMISSION, ...colours... ) ;
```

One especially useful option is:

```
glColorMaterial ( GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE )
```

This causes glColor commands to change both Ambient and Diffuse colours at the same time. That's a very common thing to want to do for real-world lighting models.

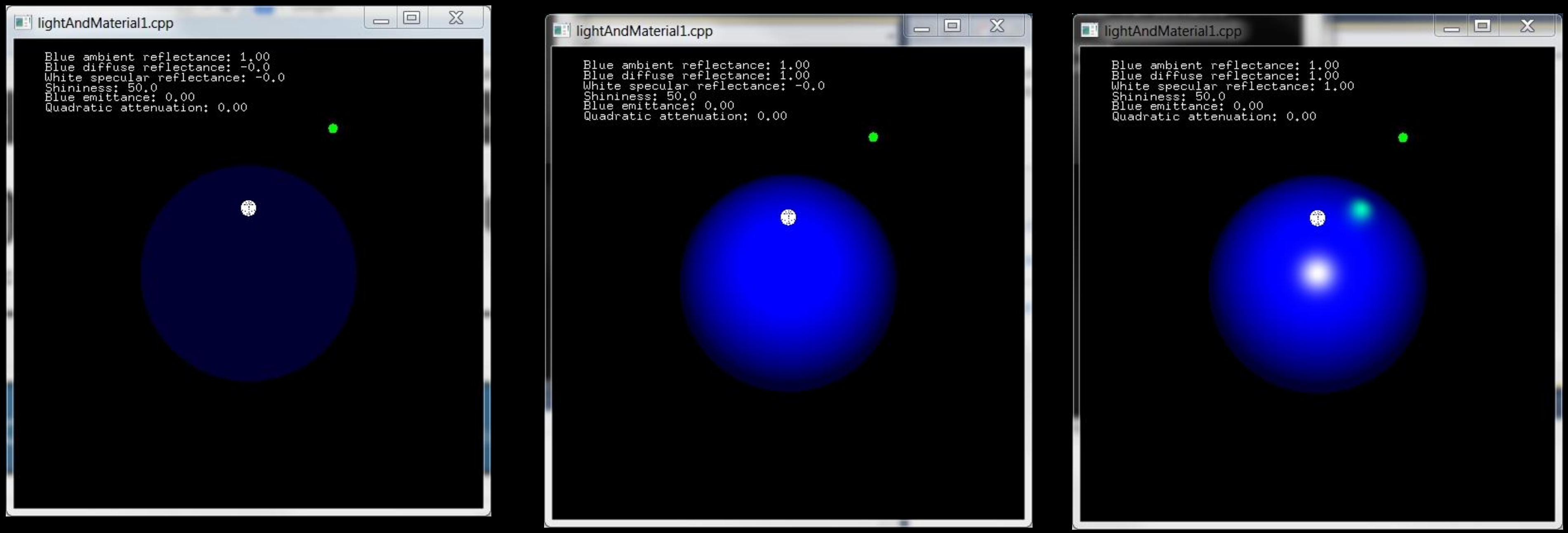
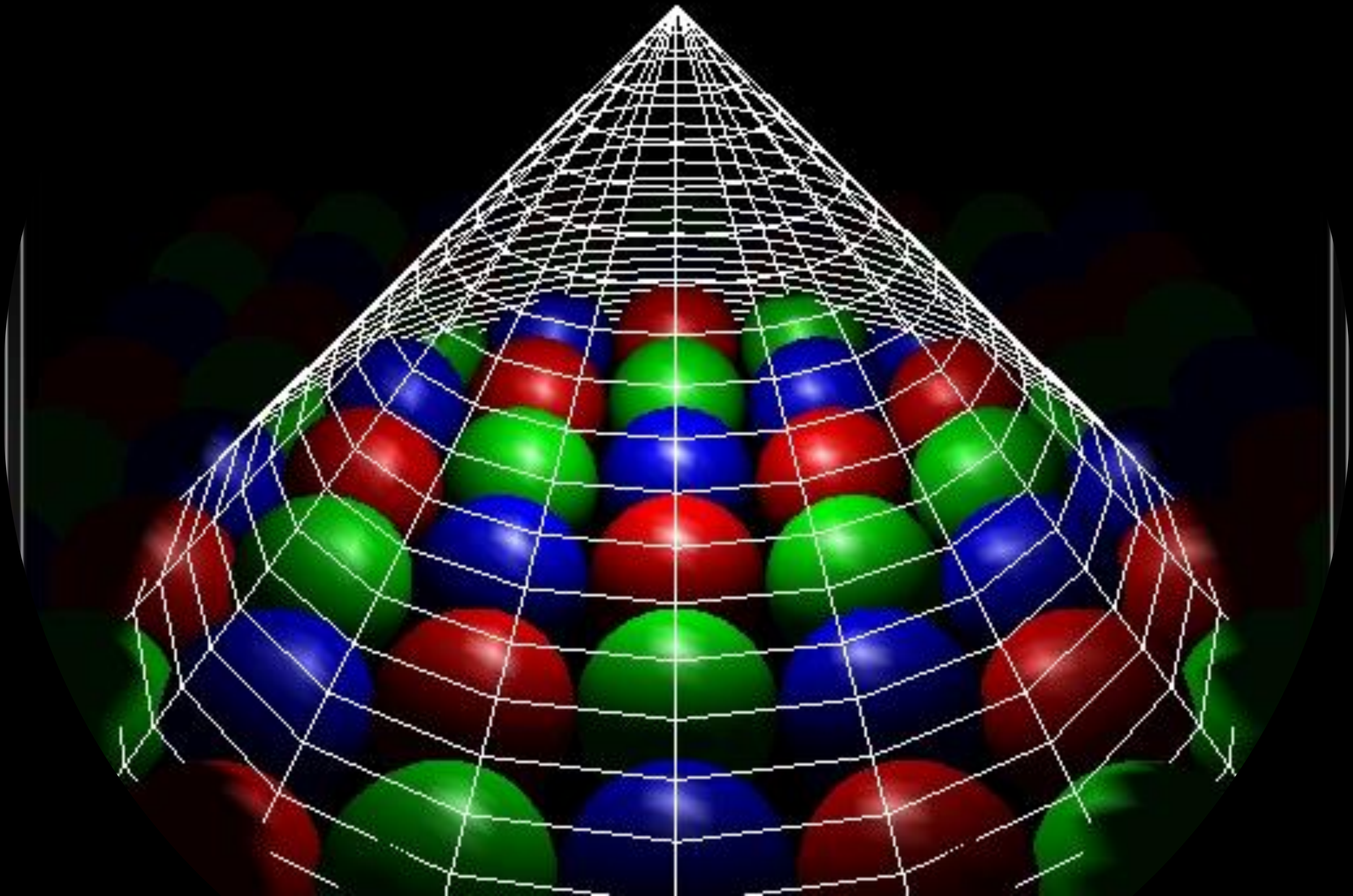


Figure 11.20: Screenshots of lightAndMaterial1.cpp: (a) Only ambient reflectance (b) Ambient and diffuse (c) Ambient, diffuse and specular.

Light attenuation: 2.00



Light Sources

OpenGL's lights are turned on and off with glEnable(GL_LIGHT_n) and glDisable(GL_LIGHT_n) where 'n' is a number in the range zero to the maximum number of lights that this implementation supports (typically eight).

The glLight call allows you to specify the colour (ambient, diffuse and specular), position, direction, beam width and attenuation rate for each light. By default, it is assumed that both the light and the viewer are effectively infinitely far from the object being lit.

You can change that with the glLightModel call - but doing so is likely to slow down your program - so don't do it unless you have to. glLightModel also allows you to set a global ambient lighting level that's independent of the other OpenGL light sources.

There is also an option to light the front and back faces of your polygons differently. That is also likely to slow your program down - so don't do it.

FURTHER READING

http://www.tomdalling.com/blog/modern-opengl/06-diffuse-point-lighting/

http://www.falloutsoftware.com/tutorials/gl/gl8.htm

http://www.mbsoftworks.sk/index.php?page=tutorials&series=1&tutorial=11

http://www.codemiles.com/c-opengl-examples/simple-light-t7304.html

https://www.glprogramming.com/red/chapter05.html