Lighting in Vizard

Excerpt from "Vizard Teacher in a Book"

LIGHTING

In order for 3D models to be visible, you need light. The easiest way to get a grasp on how virtual light works is to keep in mind how light works in the physical world. First off, light sources project rays of light. Each of these rays travels in a specific direction and varies in its color and intensity. When these rays reach an object, they bounce off the surfaces of that object into the world. Properties of the object's surface, such as color and shininess, work together to determine the nature of the reflected rays. These reflected rays continue to bounce around off of other objects. Eventually a few rays end up striking the surface of your retina and, behold, you see the object.

Digital simulations try to take these properties of light into account. In this section, we'll go over how to add light sources to a scene to produce a range of effects. We'll also go over various properties of object surfaces that effect the simulation of light reflected off those surfaces. See "LIGHTING" on page 59

One important distinction in light simulation is between simulating **local** and **global** illumination. Local illumination deals only with an individual object and the light

sources that effect it. Global illumination, on the other hand, attempts to account for the effects of all the objects in a scene on each other. So, global illumination simulates light bouncing off of one object onto another or the casting of shadows. While including global lighting effects makes for a more realistic simulation, calculating the many interrelationships between objects in real time is complex and costly to processing. There are, however, simple tricks to simulating global illumination that we will describe below.

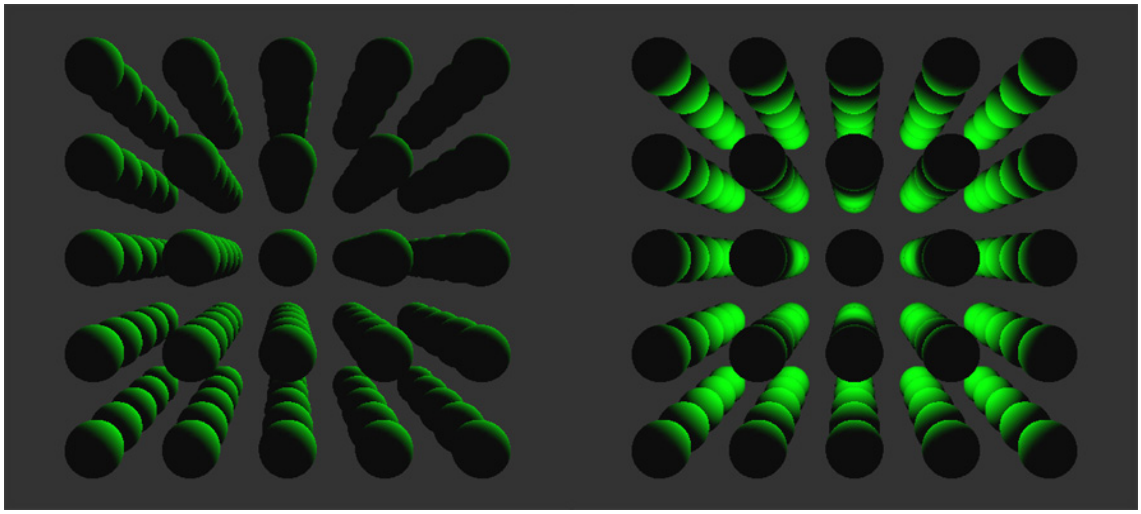
Note: when you create a world in Vizard, the program automatically adds a "headlight" to light scenes in the simulation. It's located at the viewpoint and lights the scene accordingly. In the sections below, we'll remove that headlight and explore other options for adding lights into a scene.

LIGHT SOURCES

OpenGL allows you to add up to eight light sources to a scene. Lights can be any color or intensity. One main distinction among light sources is between **directional** and **positional** lights. Directional light comes from a given direction but has no position in space. The rays from directional light sources all come in parallel from the same direction. This light source will strike all objects in the scene from this same direction. In this sense, directional light simulates light that's coming from a long distance away, like light from the sun or the moon.

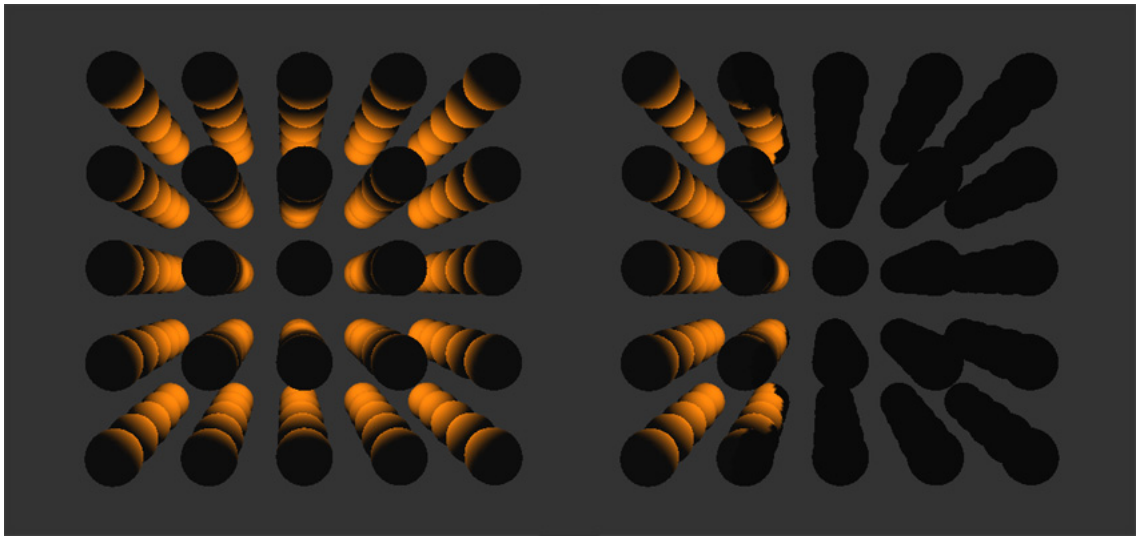
Positional lights, on the other hand, come from one point in space. The rays of light from this kind of source emanate from that one spot such that the bottom of objects above the source will be lit, the top of objects below the source will be lit, the left side of objects to the right of sources will be lit, and etc.

To get a better idea of the difference between directional and positional lights, run the light source demo. When it starts, go to "positional or directional" on the pop-down menu and try flipping back and forth between the two options. Notice how each sphere is illuminated in the same way with directional lighting but that with positional lighting, a sphere's illumination depends upon it's location with regards to the light source. (In this demo the light source is located in the middle of the cube of spheres.)



When you're dealing with positional lights, you have a number of different options available to you. You can make the source emanate light in all directions (a **point light**) or you can narrow it's focus and point it in a specific direction (a **spotlight**). With positional lights, you can also change the way in which light attenuates over distance. This attenuation can be based on a variety of formulas. If you go to the "attenuation" drop-down in the light source demo, you can change the attenuation from none to quadratic attenuation (one of the standard attenuation options).

Because spotlights have a narrow range of illumination, they have a number of adjustable parameters including the spread of the spotlight, the direction in which it points, and the degree to which the intensity of light decreases as you move away from the center of the spot. The direction and spread in the light demo are fixed, but you can vary the spot exponent by selecting different values under the "spot exponent" drop-down.



LIGHTING AND SURFACES

Intuitively we tend to think of lighting as coming from a source-- a light bulb, the sun, etc., but when modelling a virtual world it's also important to consider the relationship between light and surfaces that it bounces off. Four important categories of surfaces effects are diffuse light, specular light, ambient light, and emissive light.

Diffuse light comes from a specific direction. It illuminates the surfaces that face that direction and does not illuminate the surfaces that don't (and illuminates with decreasing intensity in between). Diffuse light reflects off a surface in all directions equally so that the lighting doesn't change as the viewing angle changes. This kind of light provides shading that makes models look 3-dimensional. Diffuse light needs a source (as discussed in the previous section) to define the direction from which the light is coming. The previous section used diffuse lighting in its demo (light source demo).

Specular light is similar to diffuse light in that its illumination depends upon the direction of the incoming rays on the surface. Specular light, however, does not reflect equally in all directions. Instead, the intensity of specular light varies as the angle of view changes. In practical terms, specular light is important because it gives glossy surfaces their shininess.

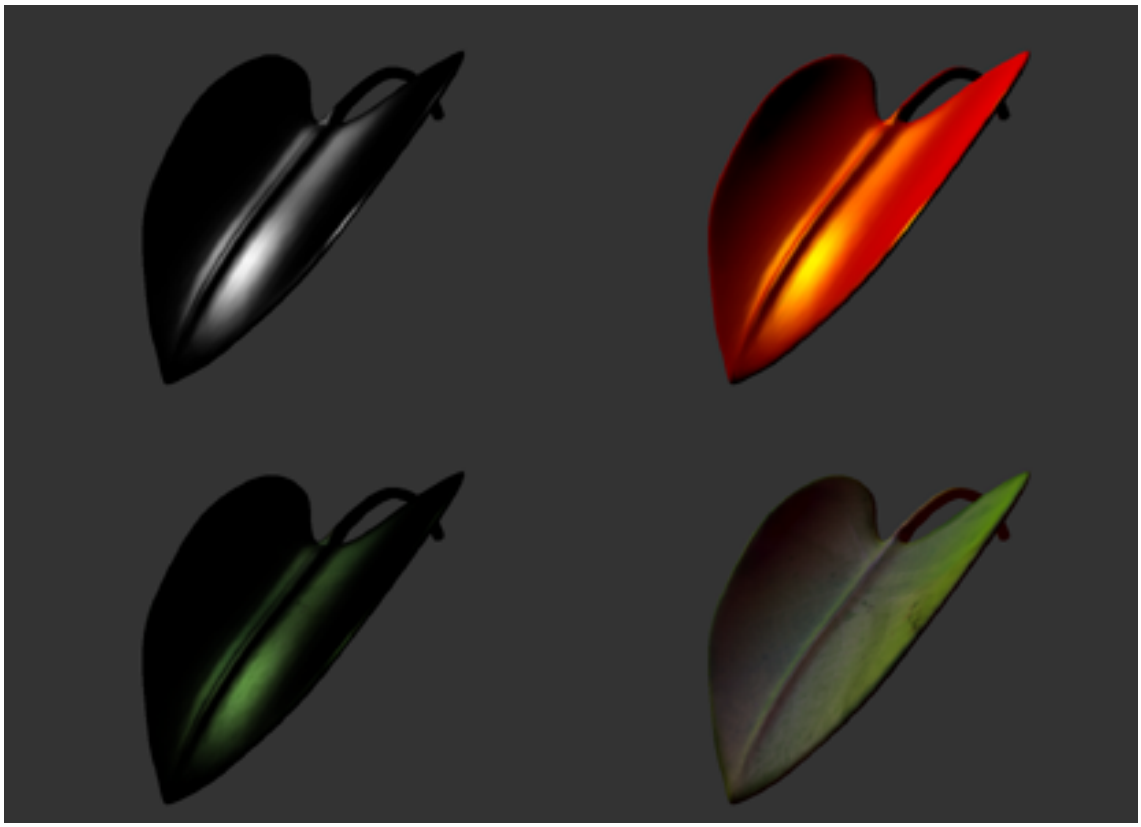
To check out the effects of specular light, run the light surfaces demo. Go to the drop-down menus at the top of the screen and go to the "specular" drop-down and choose a color for specular highlights. Then, go to the "shininess" drop-down and pick "128". This setting affects the sharpness of the specular highlights. Now rotate the viewpoint with the mouse. You should see your specular highlights move over the surface of the

leaf as you rotate the view. If you have trouble seeing the highlights, try removing the leaf's texture with the 'texture' drop-down.

Ambient light is a feature of an object's surface that emulates light coming from all directions and is scattered in all directions. Because ambient light doesn't come from any one direction, it needs no source.

To play a little bit with ambient light, run the light surfaces demo again and try changing the color of the object's ambient light with the "ambient" drop-down menu.

The fourth main kind of light is **emissive light**. An emissive light simulates light emanating from an object. Emissive illumination, light comes from all surfaces uniformly. Emissive light is a good way of simulating objects that are, themselves, simulating light sources (e.g. a light bulb). Keep in mind, though, that we're still only talking about local lighting here. So, although changing the emissive property of an object might make the object appear to glow, the light it appears to produce will not illuminate neighboring objects.



Now play a with emissive lighting by playing with the "emissive" drop-down menu. As you try different settings in this demo, note that these surface light parameters work together in the rendering of an object. So, keep in mind that the end result is a combination of all of these factors.

The kinds of lighting we've gone over in this section can be changed dynamically in a scene, but these are also aspects of an object's material (in addition to its texture). 3D modelling programs (Maya, 3DS Max, etc.) have a host of tools that allow you to manipulate these aspects of a model's material. When the object is rendered in a simulation, these components of a material will be combined with the effect of whatever light sources are in the scene.

EXAMPLE: LIGHTING A SCENE

In this example, we will use a variety of different kinds of lights to illuminate a scene.

First, let's open a Vizard script and add some models to the world and position the main viewpoint so that we can see them.

```
import viz
viz.go()

#Add a model of a forest.
forest = viz.add('art/forest.ive' )

#Add an avatar to stand there idly.
a = viz.addAvatar( 'vcc_male.cfg')
a.setEuler( [20,0,0] )
a.setPosition( [-.78,0,.3] )
a.state(1)
#Set the viewpoint's position and
#orientation so that we'll be able to see our scene.
viz.MainView.setPosition( [-.75,1.8,4.2 ] )
viz.MainView.setEuler( [-180,4, 0] )
```

Vizard has a default headlight that is linked to the main viewpoint. If you want to see what the world will look like with that head light, run your script now. After you've checked that out, disable this light so that we can see the effects of adding different kinds of lights. We do this by grabbing the main view's headlight with the viewpoint library's "getHeadLight" command. Once we have the light, we disable it with the node3d:light command "disable".

```
#Disable the default head light.
viz.MainView.getHeadLight().disable()
```

Now let's add a directional light to create the appearance of moonlight in the world. Note the arguments we use in the position command. That fourth number (0) makes the light directional. If we put a 1 there, the light will be positional. The other three numbers define what direction that light will be coming from in [x,y,z] coordinates. Since we set the y to 1 and the x and z to 0, the light will come from straight above.

```
#Add a directional light source.
moon_light = viz.addLight()
moon_light.position(0,1,0,0)
#Give the light a moonish color
#and intensity.
moon_light.color( [.6,.7,.9] )
moon_light.intensity( 1 )
```

Notice how this directional light illuminates all the upward facing surfaces. Play around with tweaking the intensity and color of the light with the lines we just added. When you're done, let's disable this light:

```
#Disable the moon light for a moment.
moon_light.disable()
```

We will attach the next light source to an object so that the object itself appears to be the light source. To help with this effect, we'll also add some emissive light to a portion of our model.

```
#Add a model of a lantern and place
#it so that it appears to hang on a the tree.
lantern = viz.add('art/lantern.ive')
lantern_position = [ 0.14 , 1.5 , 0.5 ]
lantern.setPosition( lantern_position )

#Add a light source to put inside the lantern.
lantern_light = viz.addLight()

#Define the light as a point,
#positional light. This is done
#with the last '1' in this command's
#arguments.
lantern_light.position( 0,0,0,1 )

#Link the light to the lantern.
viz.link( lantern, lantern_light )

#Grab the flame part of the lantern model
#and give an emissive quality to emulate light.
flame = lantern.getChild( 'flame' )
flame.emissive( viz.YELLOW )

#Play with the light source's parameters.
lantern_light.color( viz.YELLOW )
lantern_light.quadraticattenuation( 1 )
lantern_light.intensity( 8 )

#Give the lantern some shine.
lantern.specular(viz.YELLOW)
lantern.shininess(10)
```

Now run the script and see how it looks. To explore the effects we covered above, try tweaking the attenuation factor of the light source or the specular highlights of the lantern and run the script again. Once you're done, disable the lantern with the following line:

```
#Disable the lantern light.
lantern_light.disable()
```

Now we'll add a spotlight to the scene, linking it to a torch and animating the torch spinning.

```
#Add a model of a torch and place it in the scene.
torch = viz.add('art/flashlight.IVE')
torch.setPosition( [ -1.16 , 1.78 , 1.63 ])
#Add a light for the torch.
flash_light = viz.addLight()
#Make the light positional.
flash_light.position(0,0,0,1)
#Make this positional light a spot light by
#limiting its spread.
```

```
flash_light.spread(45)
flash_light.spotexponent( 40 )

#Link the light source to the torch.
viz.link( torch, flash_light )

torch.addAction( vizact.spin( 0,1,0,90, viz.FOREVER ) )
```

EXERCISES

1. Remove all the lights in the "Lighting a scene" example script and then add a spot-light that shines down on the avatar like a light from a helicopter.

2. Add a sphere to a world ("art/white_ball.wrl") and vary its diffuse, specular, ambient, and emissive parameters (using the <node3d>.color, <node3d>.specular, <node3d>.shininess, <node3d>.ambient, and <node3d>.emissive). Try to make it look like a bowling ball, a sun, and a cherry).