

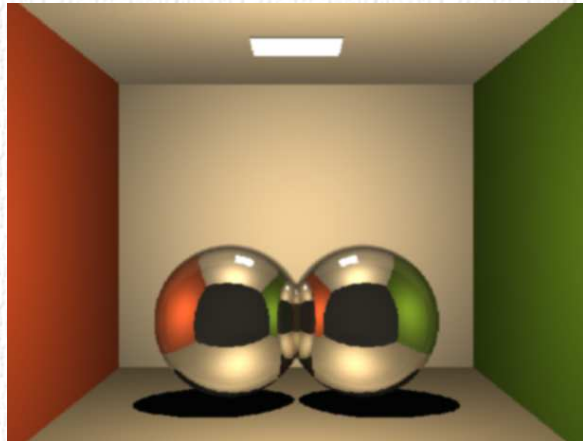
Lab 5

Kostas Vardis

<http://graphics.cs.aueb.gr/graphics/people.html>

Athens University of Economics and Business

Computer Graphics BSc



Introduction

- Blending
- Lighting, shading and Materials

Blending

- Use alpha value for transparency [0, 1]
- d parameter in the obj material (.mtl) file
(1.0 opaque, 0.0 transparent)

$$C = C_s * a_s + C_d * a_d = C_s * a_s + C_d * (1 - a_s)$$

Example:

```
// draw opaque objects
```

```
glEnable(GL_BLEND);
```

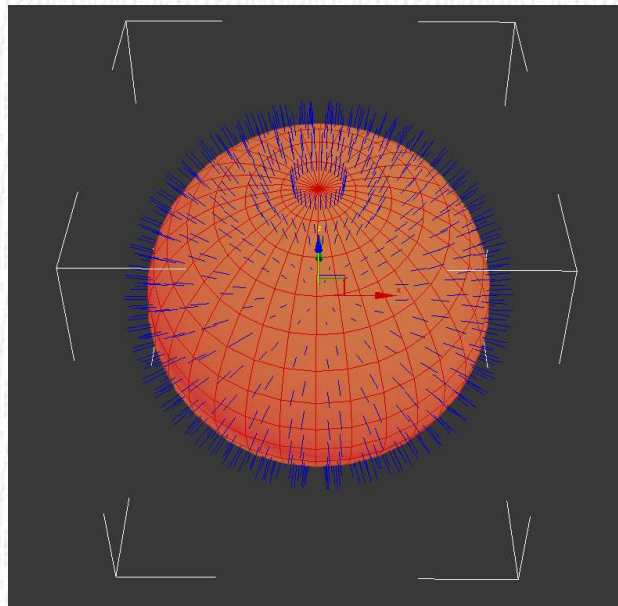
```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

```
// draw transparent objects (in back to front order!)
```

```
glDisable(GL_BLEND);
```

Normals

- Unit vector
- Perpendicular to the surface at a specific point
- Necessary for lighting calculations
- Commonly, part of vertex attributes



Lighting

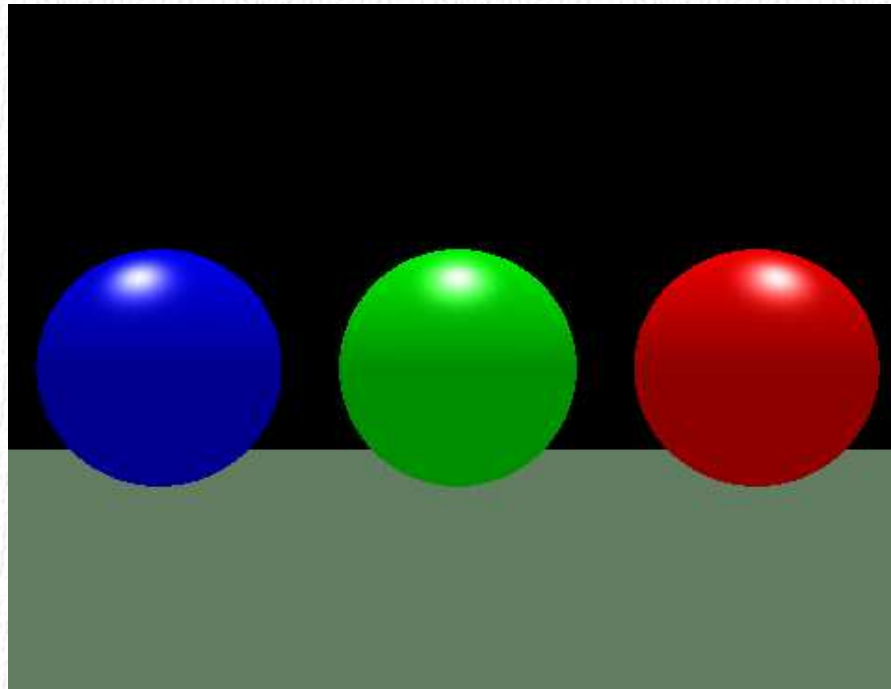
- Need to model interaction of light with a surface
- Distinguish between materials and light sources

Lights

- Directional Lights
- Omni-directional Lights
- Spotlights

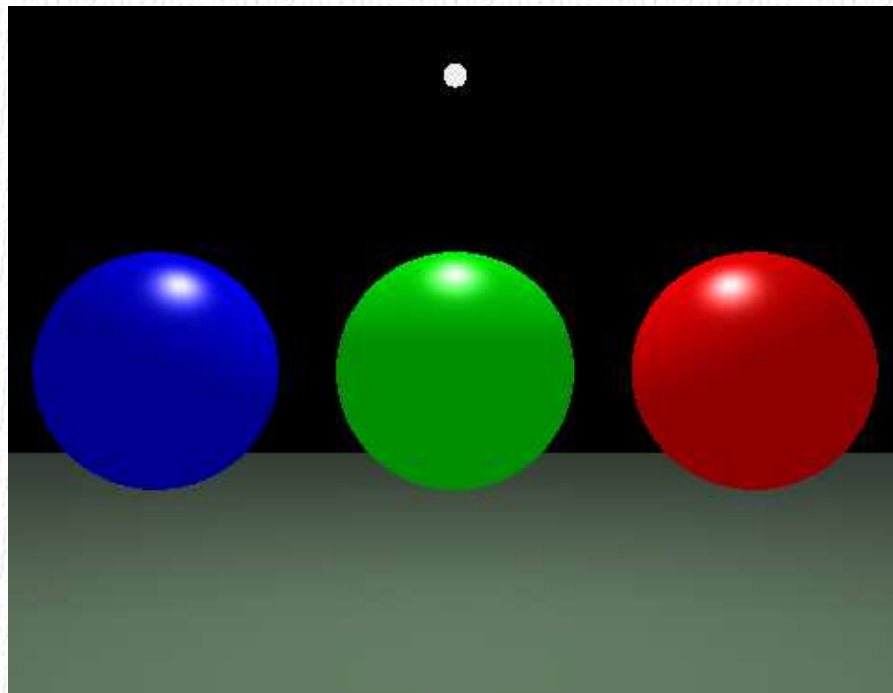
Directional Lights

- Have intensity and direction (w coordinate is zero)
- Good for distant light sources



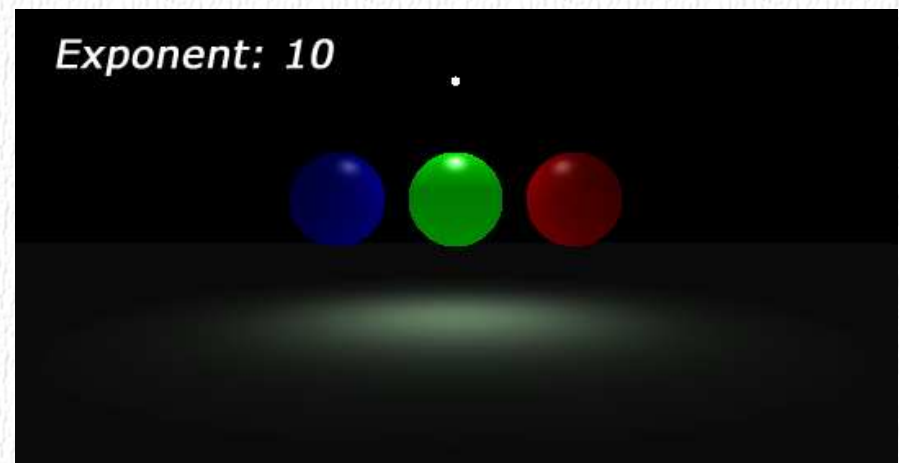
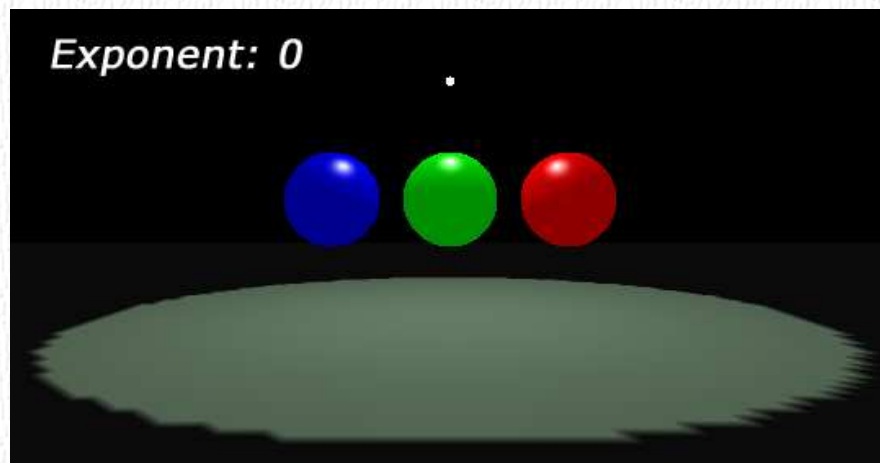
Omni Lights

- Have intensity and position (w coordinate is 1)
- Good for local light sources with no direction (e.g. candles)



Spotlights

- Point lights with intensity, position and direction
- Three parameters (cutoff angle, spotlight direction, spotlight exponent)



Attenuation

- Point lights attenuate with distance
- Apply the attenuation factor in the fragment shader

$$attenuationFactor = \frac{1}{constantAtt + dist * linearAtt + dist^2 * quadraticAtt}$$

Shading

- An object is associated with materials
- Ambient (indirect light)
- Diffuse (light scattered/absorbed under the surface)
- Specular (light reflected at the surface) and shininess (smoothness parameter)
- Blinn-Phong shading model
- Implemented in the fragment shader

Shading - ambient

For multiple lights, it should be a separate pass

$$L_{amb} = K_d * L_{constant_ambient_color}$$

Shading - diffuse

Do this once for each light

$$L_{dif} = K_d * L_{color} * \max(0.0, ndotl)$$

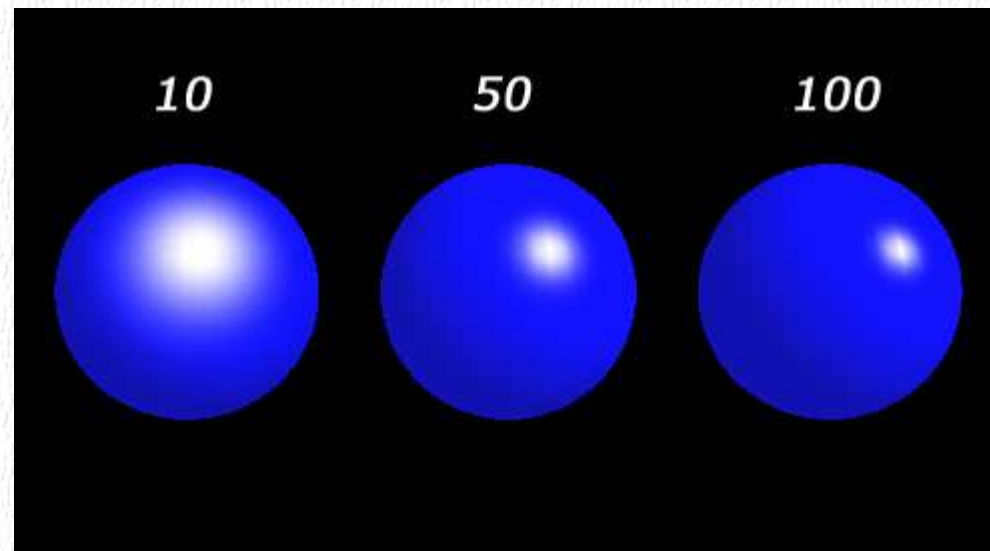
Shading - specular

Do this once for each light

$$L_{spec} = K_s * \cos h^m * L_{color} * \max(0.0, ndotl)$$

Shading - specular

- Smoothness parameter
- A smoother surface has tighter highlights



Shading - final

Final lighting is

$$L_{total} = L_{amb} + (L_{dif} + L_{spec}) * attenuation * spot$$

For multiple lights enable additive blending:

```
glEnable(GL_BLEND);
```

```
glBlendFunc(GL_ONE, GL_ONE);
```

```
// draw scene for each light (e.g. in a for loop)
```

```
glDisable(GL_BLEND);
```

Shading



Done!

Check lab5 project