EREN ÇAĞLAROĞLU 11500168978

CMPE 360 HOMEWORK 4

PART 1

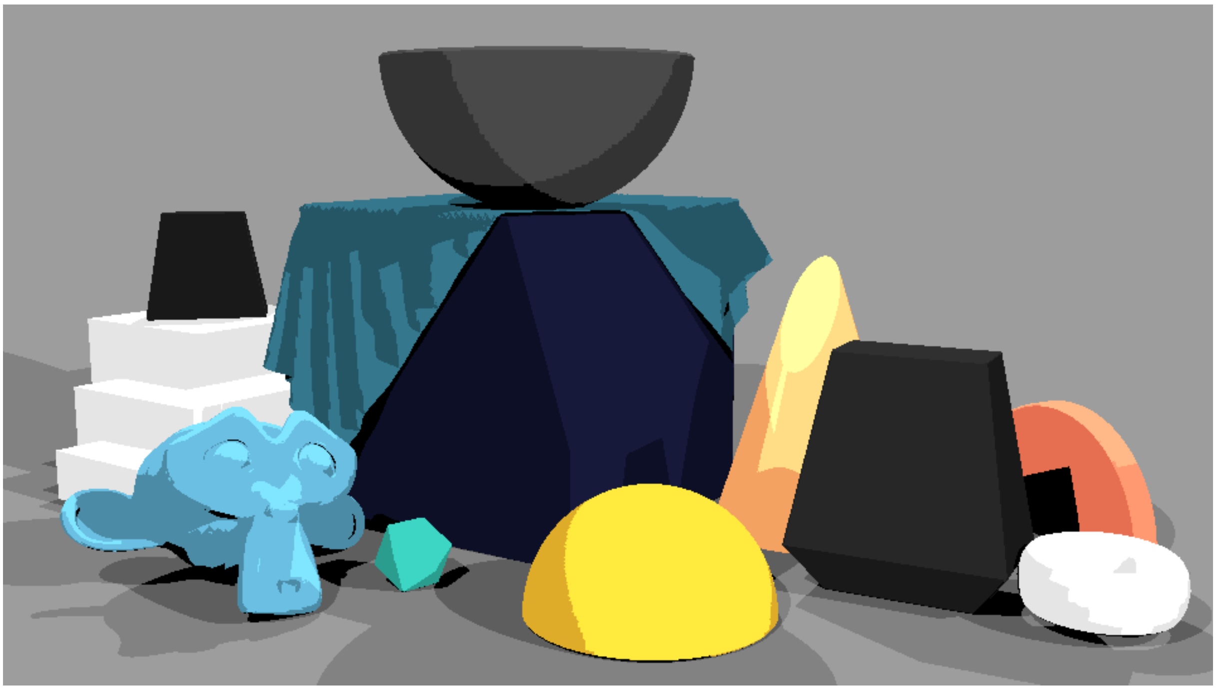

Figure 1 Shadow Ray.

Shadow Ray

1.Calculate light_vec: light_vec indicates the vector that goes from the light source's position (light.location) to the hit location (hit_loc). The hit location is subtracted from the light location to calculate it. This vector points in the direction of the light source from the spot on the object's surface.

2.Normalize light_vec: By normalizing light_vec, the light_dir vector is obtained. Normalization maintains the direction of the vector while guaranteeing that its length is 1. This is required to determine the shadow ray's direction.

3.Compute new_orig: The shadow ray's origin is represented by the new_orig variable. The hit position (hit_loc) has a slight offset (epsilon) applied to it in the direction of the light source to avoid self-intersection and numerical problems. To minimize self-shadowing artifacts, this offset makes sure that the shadow ray begins somewhat above the surface rather than directly from it.

4.Cast the shadow ray: Blender's ray_cast function is used to cast the shadow ray. It determines if any objects in the scene are intersected by the ray that originates at new_orig and travels in the direction of light_dir.

5.check for has_light_hit: This variable stores a boolean value indicating if the shadow ray contacted any objects in the scene. The point is in shadow if has_light_hit is True because the shadow ray passed through an item before meeting the light source. In this instance, the code proceeds to the next light source, ignoring the current one.

This part main goal is to ascertain whether a certain light source is illuminating a spot on the object's surface. When the shadow ray meets an item before it reaches the light source, it indicates that the point is in shadow and that the light source at that location is not contributing to the final color. The shading calculations can move forward if the shadow ray passes through an object-free area, indicating that the point is illuminated by the light source.

PART 2

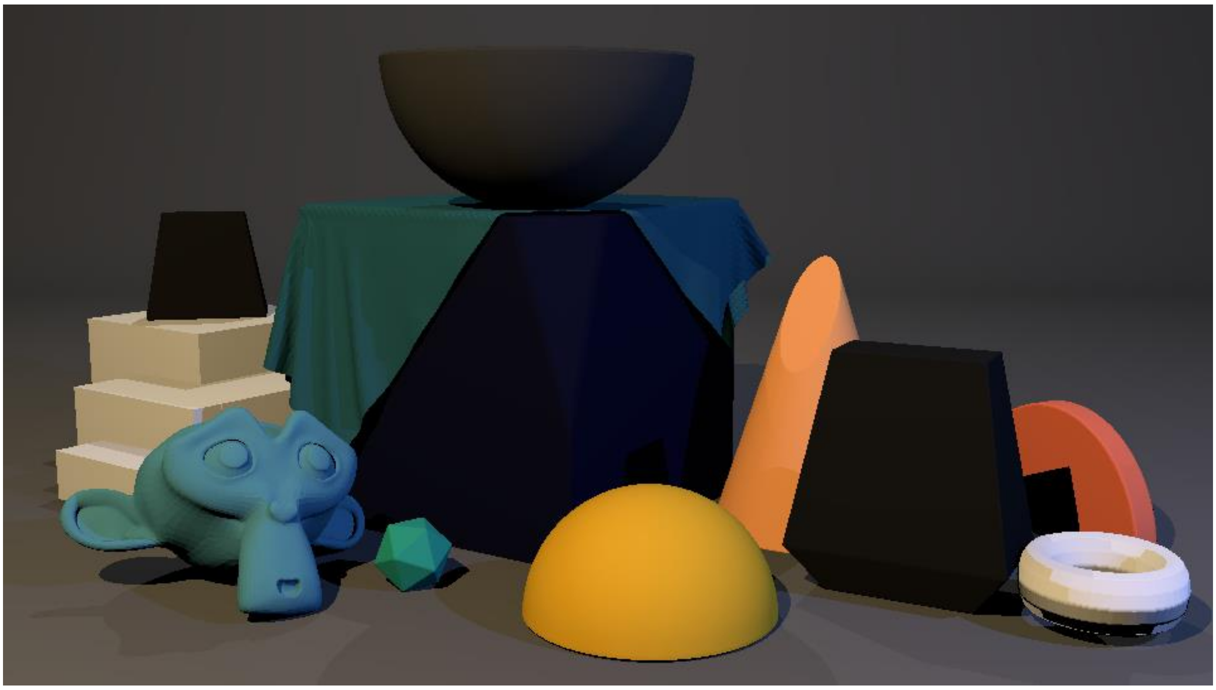

Figure 2 Blinn-Phong diffuse-shading

Figure 3 Blinn-Phong specular-shading

Diffuse Reflection Component:

Using the variable" I_light", the code determines the light's intensity at the spot. The inverse-square law, which accounts for the distance between the point and the light source, attenuates this intensity, which is dictated by the hue of the light source (light_color). An increased intensity is produced at a closer range."light_dir" is a representation of the direction from the surface point to the light source. This is computed using the normalized vector from the point to the position of the light source (previously computed as "light_vec").

The Lambertian reflection model, which is dependent on the angle between the normalized "light_dir" and the surface normal ("hit_norm"), is used by the code to compute the diffuse reflection component.

Specular Reflection Component:

The Blinn-Phong reflection model is used to compute the specular component, simulating bright spots on the surface.The computation makes use of the light's intensity (I_light) once more.The halfway vector, or half_vector, between the vision direction (view_dir) and the light direction (light_dir), is computed by the code. The direction that is exactly between the view direction and the light direction is represented by the midway vector.

The specular reflection is computed as the raised to the power of specular_hardness dot product of the half vector and the surface normal (hit_norm). The specular variable holds the outcome.

PART 3

Figure 4 Ambient Light

Ambient Light

The rendering process' ambient component is considered. All the items in the scene are impacted by the ambient lighting, which is consistent and homogeneous regardless of how the objects are positioned or oriented in reference to the light sources.

It determines if the spot on the surface—indicated by no_light_hit—is not directly lit by any of the light sources in the scene.The diffuse color of the material (k_diffuse) and the scene's ambient color (k_ambient) are multiplied to determine the ambient component I_ambient if the point is not directly lighted.When the pixel's color is increased by I_ambient, it guarantees that the point gets some lighting even in the absence of direct light from sources.

In conclusion, this method guarantees that every point on the surface receives a minimal amount of illumination, which improves the image's overall visibility and realism and gives ambient color control over the image's mood and atmosphere.