EREN ÇAĞLAROĞLU 1500168978
DENİZ CANER AKDENİZ 14021504096

CMPE 360 HOMEWORK 5

PART1

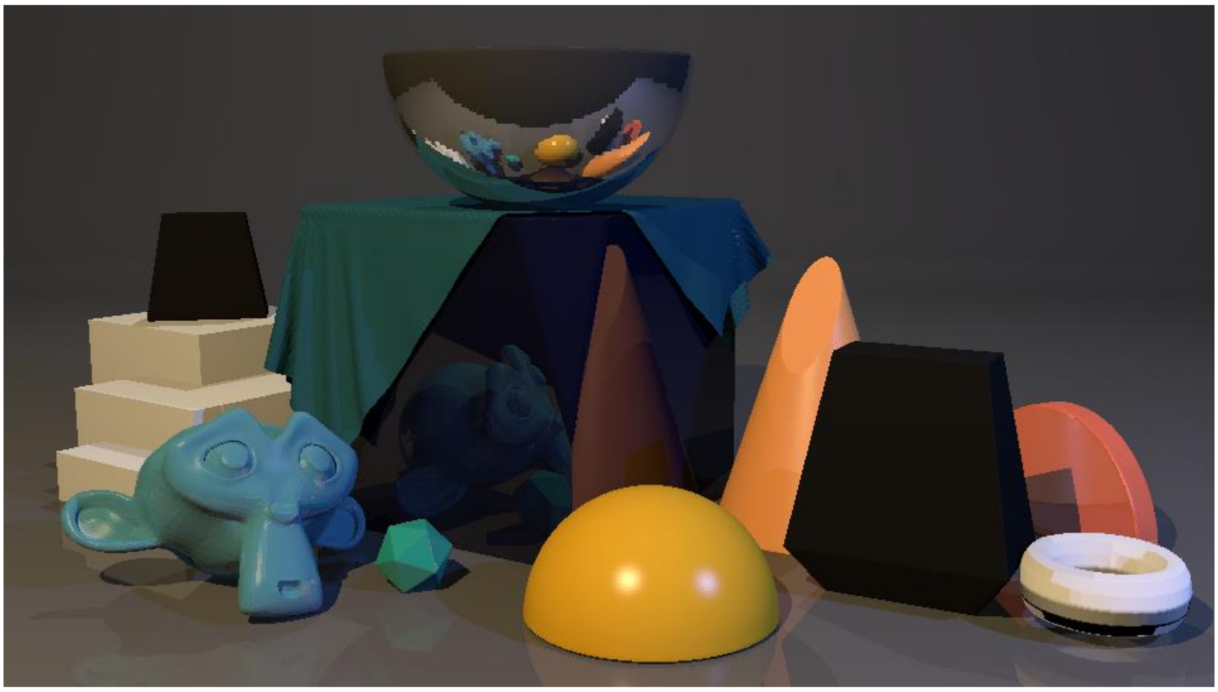

Figure 1 TODO 4: RECURSION AND REFLECTION

- **Calculate Reflection Ray Direction:**

  Calculates the direction of the reflection ray based on the incident ray direction and the surface normal at the point of intersection.

- **Set up the Reflection Ray:**

  Defines the origin of the reflection ray, slightly offset from the intersection point to avoid self-occlusion.

- **Recursively Trace the Reflection Ray:**

  Recursively traces the reflection ray using the RT_trace ray function. The returned color represents the contribution of the reflected ray.

- **Combine Reflection Color with Pixel Color:**

  Multiplies the reflection color by the material's mirror reflectivity and adds it to the current pixel color.

EREN ÇAĞLAROĞLU 1500168978
DENİZ CANER AKDENİZ 14021504096

- **Update Depth:**

  Decreases the recursion depth. It's essential to keep track of the depth to prevent infinite recursion.

- **Summary**

  This code segment employs ray tracing to compute reflections. It's part of a larger ray tracing application that includes interactions like reflections, shadows, and material properties.

PART2

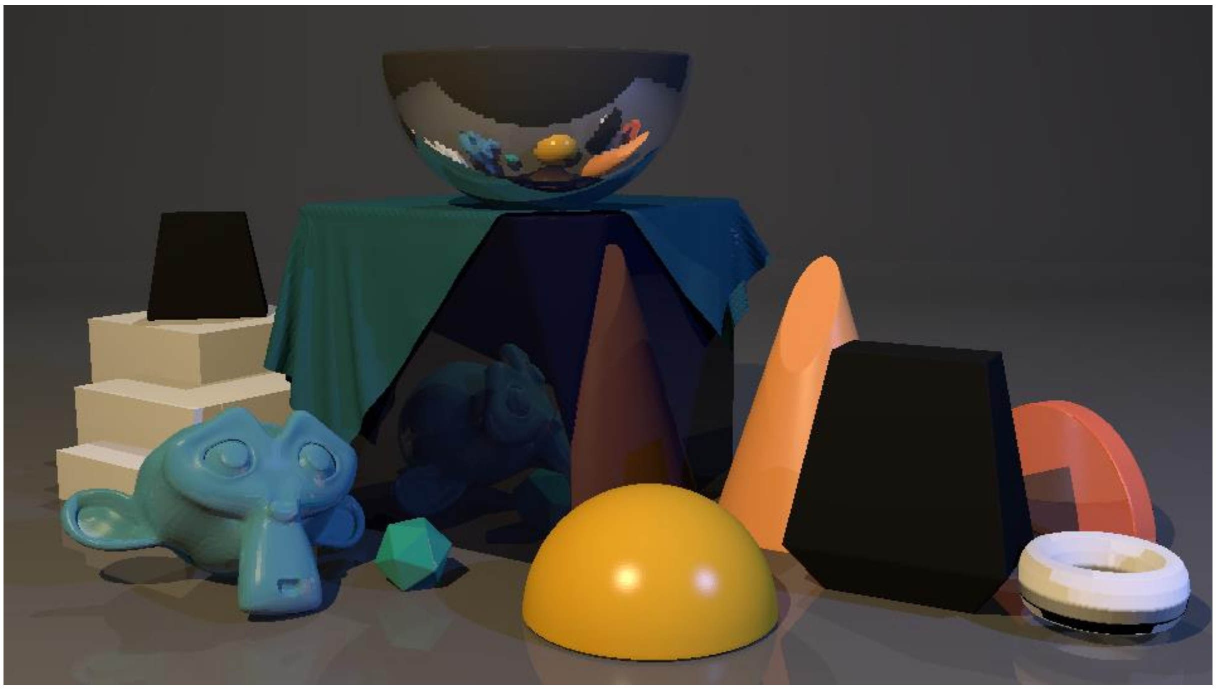

Figure 2 TODO 5: FRESNEL

- **Direct Reflectivity or Fresnel Calculation:**

  If Fresnel is not used (mat.use_fresnel is False), the reflectivity (reflectivity) is directly taken from the material's mirror reflectivity property.

- **Fresnel Reflectivity Calculation:**

  If Fresnel is enabled (mat.use_fresnel is True), it calculates the Fresnel reflectivity using Schlick's approximation. This involves determining the Fresnel reflectance at normal incidence (R_0) and then adjusting it based on the incident angle (cos_theta). The final reflectivity is used for subsequent calculations in ray tracing.

- **Summary**

  This code decides whether to use Fresnel reflection. If enabled, it calculates the reflectivity with Schlick's approximation, considering the index of refraction of the material (mat.ior). The calculated reflectivity is then used in the ray tracing process.

PART3

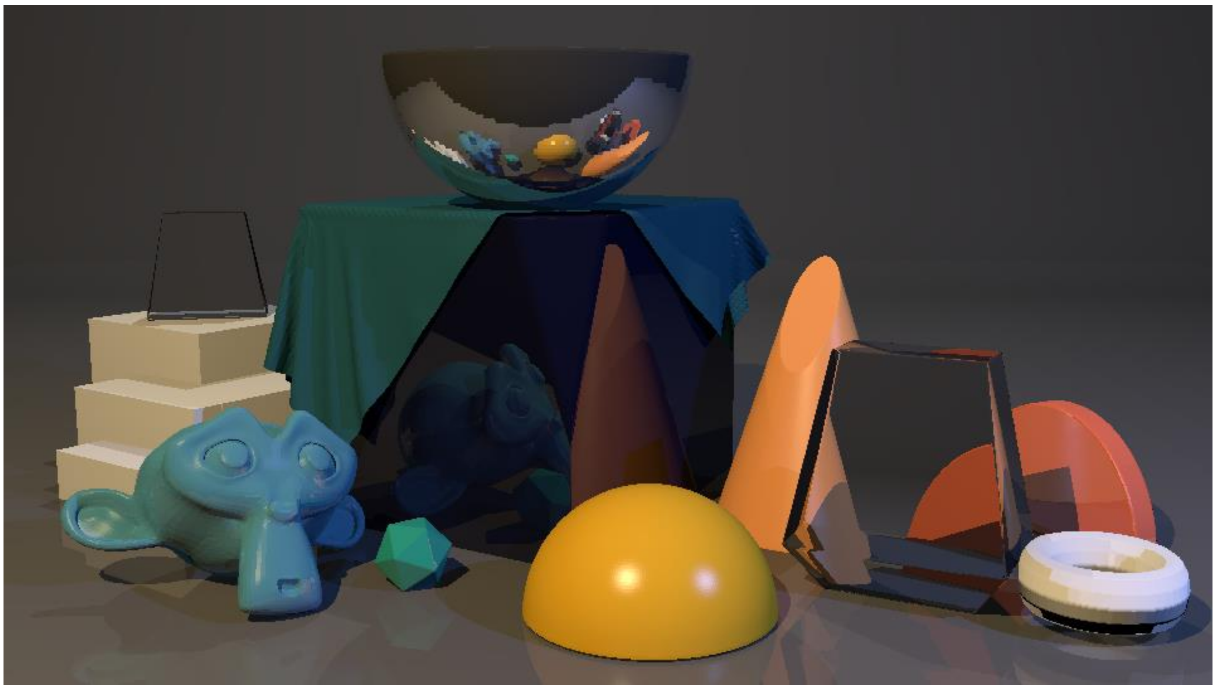

Figure 3 TODO 6: TRANSMISSION

- **Check for Transmission:**

  It checks if the material has transmission properties (mat.transmission > 0). If so, it means the material is transparent, allowing for refraction.

- **Calculate Refraction Parameters:**

  Calculates the cosine of the angle between the ray direction and the surface normal (cos_theta).

  Computes the term under the square root (sqrt_term) in the refraction formula.

- **Check for Valid Refraction:**

  Checks if the value under the square root is non-negative (sqrt_term >= 0). If negative, there is total internal reflection, and no transmitted ray is generated.

- **Generate Transmitted Ray:**

  Computes the normalized surface normal (hit_norm_norm) and the new origin for the transmitted ray.

  Calculates the direction of the transmitted ray (D_transmit) using the refraction formula.

- **Recursively Trace Transmitted Ray:**

  Calls the RT_trace_ray function to recursively trace the transmitted ray, obtaining the color contribution (L_transmit).

- **Update Pixel Color:**

  Adds the color contribution of the transmitted ray to the overall pixel color. The contribution is scaled by (1 - reflectivity) * mat.transmission, considering both transmission and the reflectivity calculated earlier.

- **Summary**

  This code segment handles the transmission of rays through transparent materials, considering refraction effects. The transmission color contribution is calculated and added to the overall pixel color.

PART4

- Reflection (TODO4) and Fresnel Reflection (TODO5) both handle reflection effects. We think TODO5 has more realistic reflection when we zoom the images. Example when we zoom the reflection, we think the reflections on the ground are more noticeable.
- In TODO6, we transformed the surfaces into a glassier structure, making it look like a permeable structure. Unlike other TODOs, its structure is simpler and more permeable.