

ETGG1803

Lab8: Ray intersections and checker material

Points: 13

Assigned: 2/10/2010 (Section 03), 2/17/2010 (Section 02), 2/18/2010 (Section 01)

Due: 2/23/2010 (Tuesday) by 11:59pm

Tasks:

1. (You can use the Lab08Start.zip file on the web page if you don't have the class code we did together)
2. **(5 points)** Complete the sphere's hitTest method.
 - a. Use the notes I gave you in class and/or section 13.12 of the book to determine if we have a hit with the Ray that was passed to us and this sphere (self)
 - i. If it doesn't hit, return None
 - ii. If it does, create and return a new HitInfo object.
3. **(3.5 points)** Complete the plane's hitTest method.
 - a. The goal is similar to that of the sphere's:
 - b. Use your notes from class and/or section 13.9 in the book to make the determination.
 - c. Ensure that you return None in all of these cases:
 - i. The ray's direction is parallel to the plane.
 - ii. The ray originates on the "back" side of the plane.
 - iii. The ray points away from the plane.
4. **(1.5 point)** Modify the HitInfo class to contain a Normal vector.
 - a. Make sure this is normalized.
 - b. The normal you supply will depend on the type of object hit, and (for the sphere), the position of the hit point.
 - c. You aren't given the hit point directly, but you can generate it using the Ray and the t-value you computed in tasks 2 and 3.
5. **(3 points)** Finish the CheckerboardMaterial's getColor method.
 - a. You'll use the spacing attribute and the position passed to you to determine which of colorA and colorB you return.
 - b. Use the notes I gave you in class to make this determination.
 - c. Hint:
 - i. Calculate valX as follows¹:
 1. If $0.0 \leq \text{pos.x} < 2.0$, valX is 0
 2. If $2.0 \leq \text{pos.x} < 4.0$, valX is 1
 3. If $4.0 \leq \text{pos.x} < 6.0$, valX is 0
 4. If $-2.0 \leq \text{pos.x} < 0.0$, valX is 1.
 5. If $-4.0 \leq \text{pos.x} < -2.0$, valX is 0.
 6. And so on...
 - ii. Calculate valY and valZ similarly.
 - iii. If valZ is even, and valX equals valY, return ColorA
 - iv. If valZ is odd, and valX does NOT equal valY, return ColorA.

¹ Basically valX is telling you if the "tile" number for X is even (valX is 0) or odd (valX is 1)

v. All other cases, return ColorB.

Sample Image:

generated with:

- *A plane (normal=(0,1,0), d=0) with a checker material(spacing=1.0, colorA=(0,0,0), colorB=(1,1,1))*
- *A sphere (radius=2.5, center=(0,0,0)) with a Red solid material.*
- *Camera at (0,2,20) coi at (0,0,0)*

