Review problems

1. **Lighting:** You are given a unit sphere centered at the origin and light source specified with the following code:

   ```javascript
   var pointLight = new THREE.PointLight("white", 2, 30); // color, intensity, distance light travels
   pointLight.position.set(2, 2, 0);
   ```

   What is the unit normal vector \( \vec{n} \) at the point \( \vec{p} = \left( \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \) on the sphere? What is the unit direction of the light ray \( \vec{l} \) pointing towards this point? Compute the dot product of these two unit vectors. What does this tell us about the color of sphere at this point?

2. **Projection:** (adapted from the Fall 2015 final exam) You are given the following 8 vertices of a cube in world space, a camera at the origin, and a viewport at \( z = -1 \).

   (a) Fill in the table below with the 2D viewport coordinates for each type of projection. Assume the viewport is large enough that no points will be clipped out.

<table>
<thead>
<tr>
<th>world coordinates ( \rho_i )</th>
<th>orthographic projection</th>
<th>perspective projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1,1,-2) )</td>
<td>( (1,1) )</td>
<td>( (\frac{1}{2}, \frac{1}{2}) )</td>
</tr>
<tr>
<td>( (3,1,-2) )</td>
<td>( (3,1) )</td>
<td>( (\frac{3}{2}, 1/2) )</td>
</tr>
<tr>
<td>( (3,3,-2) )</td>
<td>( (3,3) )</td>
<td>( (3/2, 3/2) )</td>
</tr>
<tr>
<td>( (1,3,-2) )</td>
<td>( (1,3) )</td>
<td>( (1/2, 3/2) )</td>
</tr>
<tr>
<td>( (1,1,-4) )</td>
<td>( (1,1) )</td>
<td>( (1/4, 1/4) )</td>
</tr>
<tr>
<td>( (3,1,-4) )</td>
<td>( (3,1) )</td>
<td>( (3/4, 1/4) )</td>
</tr>
<tr>
<td>( (3,3,-4) )</td>
<td>( (3,3) )</td>
<td>( (3/4, 3/4) )</td>
</tr>
<tr>
<td>( (1,3,-4) )</td>
<td>( (1,3) )</td>
<td>( (1/4, 3/4) )</td>
</tr>
</tbody>
</table>
(b) Draw what the "viewer" would see in each case (only one quadrant of the viewport is shown).

Orthographic

3. Name the algorithm based on the pseudocode below:

for all p in pixels:
    create a ray r from camera to p
for all o in objects of the world:
    calculate intersection of o with r
    keep if closest
    color p based on material of o & angle of surface to light

Perspective

4. Ray tracing: (adapted from the Fall 2015 final exam) You are trying to figure out whether a circular mirror on a wall in your scene is visible from a certain pixel. The direction of the ray and the point on the viewport are, respectively:

\[ \vec{R}_d = \left( \frac{3}{5}, 0, -\frac{4}{5} \right), \quad \text{and} \quad \vec{p}_v = (-1, 2, -1) \quad (z = -1 \text{ included for clarity}) \]

(a) The camera is 5 units away from the point \( \vec{p}_v \). Where is the camera located (find \( \vec{R}_o \))?

\[
\begin{bmatrix}
-1 \\
2 \\
-1
\end{bmatrix} = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} + 5 \begin{bmatrix} 3/5 \\ 0 \\ -4/5 \end{bmatrix}
\Rightarrow
\begin{align*}
c_x &= -1 - 3 = -4 \\
c_y &= 2 - 0 = 2 \\
c_z &= -1 + 4 = 3
\end{align*}
\]

\[ \vec{R}_o = (-4, 2, 3) \]
(b) The circular mirror is located on the wall represented by the plane \( x = 5 \). What are the coordinates of \( \vec{p}_w \), the point where this ray intersects the wall? How far away is \( \vec{p}_w \) from the camera?

\[
\vec{R}(t) = \begin{bmatrix} -4 \\ 2 \\ 3 \end{bmatrix} + t \begin{bmatrix} 3/5 \\ 0 \\ -4/5 \end{bmatrix}
\]

\[
x = 5 = -4 + t \cdot \frac{3}{5} \Rightarrow t = \frac{9}{3} \Rightarrow t = 3
\]

\[
y = 2 + t \cdot 0 \Rightarrow y = 2
\]

\[
z = 3 + 15 \left( -\frac{4}{5} \right) = 3 - 12 = -9
\]

\[
\vec{p}_w = (5, 2, -9)
\]

(c) The circular mirror has center point \( \vec{c} = (5, 5, -7) \) and radius \( r = 3 \). Does this ray intersect the mirror? Justify your answer (it might be helpful to draw a picture of the wall).

Distance between \( \vec{p}_w \) and \( \vec{c} \) should be less than \( r \) if ray intersects mirror:

\[
\| \vec{p}_w - \vec{c} \| = \sqrt{(5-5)^2 + (2-5)^2 + (-9-(-7))^2}
\]

\[
= \sqrt{0 + 9 + 4} = \sqrt{13} = l
\]

\[\sqrt{13} > 3 = r \Rightarrow \text{no outside circle}\]

(d) Create a general algorithm for determining whether a ray intersects a circular object lying on a given plane. You don’t need to use code or pseudocode, just a general description of how to find the equivalent of the point on the plane \( \vec{p}_w \), and then an inequality in terms of \( \vec{p}_w, \vec{c}, \) and \( r \).

First, find the intersection point \( \vec{p}_w \) of the ray and the plane the circle lies on. Then, to test whether the point is inside the circle, check:

\[
\| \vec{p}_w - \vec{c} \| \leq r \rightarrow \text{if no, no intersection with circle}
\]

\[
\| \vec{p}_w - \vec{c} \| \leq r \rightarrow \text{if yes, ray does intersect circle}.
\]