Last Name: ______________ First Name: ______________ Cornell NetID: ______________

CS 4620 Midterm, March 21, 2017

This 90-minute exam has 4 questions worth a total of 100 points. Use the back of the pages if you need more space.

Academic Integrity is expected of all students of Cornell University at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare I shall not give, use, or receive unauthorized aid in this examination.

Signature: ______________ Date ______________

1. [30 points] **Meshes**

   Suppose we are given the geometry shown in Figure 1, and the following position buffer containing floating point data:

   {-1.0, -1.0, 0.0, 0.0, -1.0, 0.0, 1.0, -1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0}

   We are also given the following color buffer, which stores color values in RGB format:

   {0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0}

   Suppose we were to store a color at each vertex of each triangle in the mesh. Each pixel on a triangle could then be determined by linearly interpolating the colors at each of its vertices weighted by barycentric coordinates. Figure 2 shows an example of this technique.

   (a) Write the index list that corresponds to the mesh shown in Figures 1 and 2. Each entry of the list should be in the form i/j, where i is an index into the position buffer above, and j is an index into the color buffer above. For instance, "1/2" means a vertex with position (0.0, -1.0, 0.0) and color (0.0, 1.0, 0.0). The index list should specify the triangle faces in the order labeled in Figure 1. When multiple orderings of indices are possible, chose the ordering that puts the lowest position index first. Use 0-based indexing. The triangles should be oriented such that the outward-facing normal is pointing out of the page.

   (b) If we were to modify the position buffer above, could we change the mesh's topology? Could we change its geometry? Explain your answer.

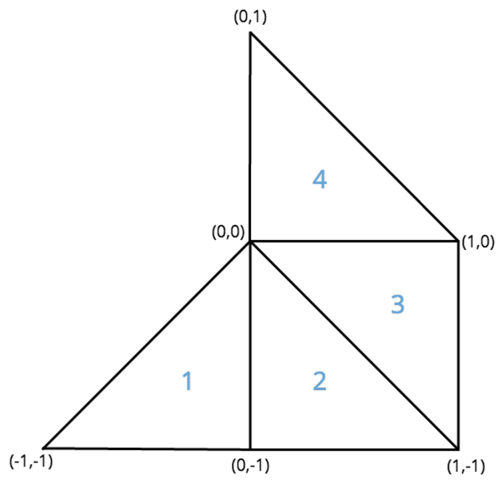


Figure 1: The target mesh configuration. All Z-coordinates of all positions are 0.

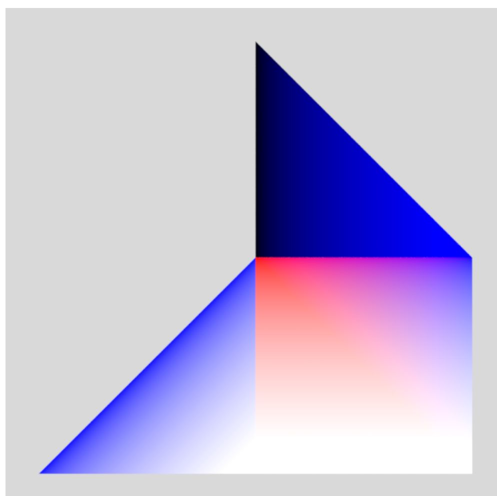


Figure 2: The target mesh, fully rendered. Color data is stored at each vertex, and is interpolated across each triangle using linear interpolation weighted by barycentric coordinates.

2. [15 points] **Texturing**

   Consider the following texture:

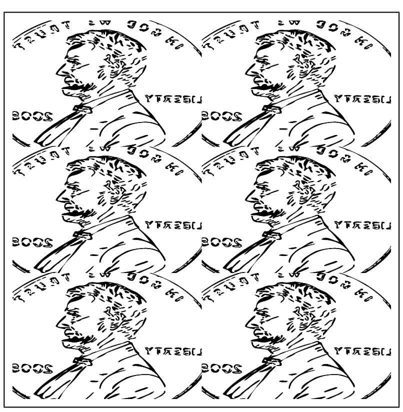
(0, 1) (1, 1)

(0, 0) (1, 0)

For each of the following images, label the four corners with UV coordinates that produce that result when the above texture is applied to a square. All of the corner UV coordinates are integers. (The borders are not part of the images.)

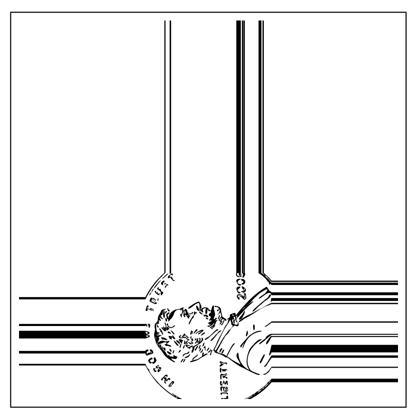
Also, write down which texture wrap mode would produce that result.

(a)



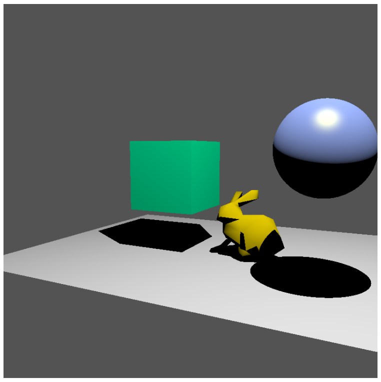
Wrap mode:

(b)

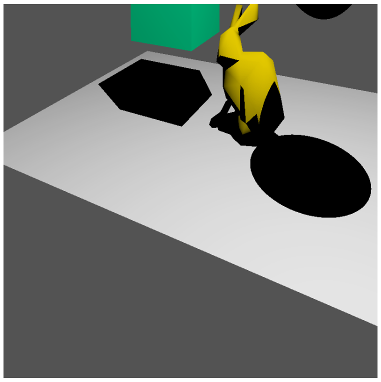


Wrap mode:

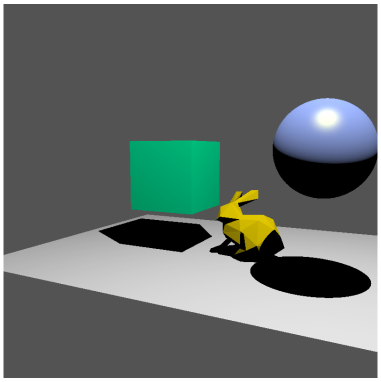
3. [25 points] **Ray Tracing Bugs**



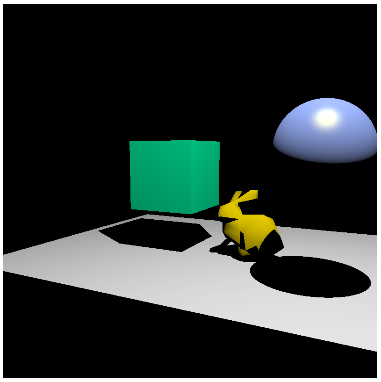
The image here is a correct ray traced rendering of a scene consisting of a single light, a box, a bunny mesh, a sphere, and a plane. The sphere has Blinn-Phong shading, and all other surfaces have Lambertian. We have modified the ray tracer to introduce several single-line bugs, one for each image you see below. For each case, match the image with the bug described on the next page, and give a brief description of how the bug leads to the visible changes in the image. Each image corresponds with exactly one bug, and the first one is completed as an example.



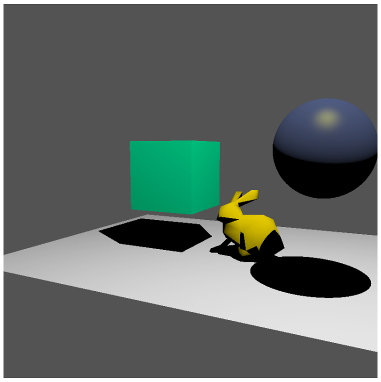
(a)



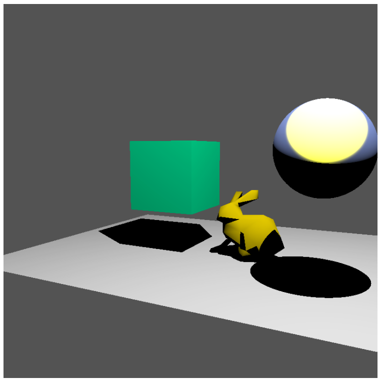
(b)



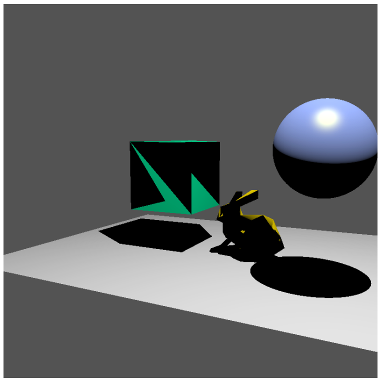
(c)



(d)



(e)



(f)

0. The triangle intersection code ignores vertex normals provided in the input mesh.

   This bug corresponds to image (b). Without vertex normals, the ray tracer is using the triangle normals, leading to a faceted appearance.

1. The half vector for Blinn-Phong shading is not normalized.

2. Rays that don't intersect any object are not handled properly.

3. The camera is generating incorrect ray directions.

4. The light intensity is not factored into Blinn-Phong shading.

5. Ray intersection is not properly determining the first object hit.

4. [30 points] **Transformations** [1]

(a) The following transformation is a rotation in 3D:

$$\begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

What is the axis of rotation? By how many degrees counterclockwise?

(b) A unit sphere $S = \{(x, y, z) : x^2 + y^2 + z^2 = 1\}$ is transformed by the following affine transformation:

$$\begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 2 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 2 \\ & 1 & & 5 \\ & & 1 & 6 \\ & & & 1 \end{bmatrix}$$

Consider the point $\mathbf{a} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ on S. Where is $\mathbf{a}$ after the transformation? What is the normal vector at $\mathbf{a}$ after the transformation?

[1] In case anyone is blanking on these: $\cos 30° = \sin 60° = \frac{\sqrt{3}}{2}$; $\cos 60° = \sin 30° = \frac{1}{2}$; $\cos 45° = \sin 45° = \frac{\sqrt{2}}{2}$.

(c) Consider the scene graph given on the left. The "Circle" and the "Square" nodes denote the shapes given on the right. Draw the transformed shapes according to the scene graph in the space provided on the following page. Great precision in drawing is not needed; if your lines are within 1/4 unit of the right location, that is sufficient.