

**CS445 Final***Fall 2012*

1. (max = 8)	5. (max = 8)
2. (max = 12)	6. (max = 14)
3. (max = 20)	7. (max = 17)
4. (max = 9)	8. (max = 12)
Final Score: (max=100)	

1) (8 pts) **Rays**

What is the parametric equation of a ray? Besides giving the formula, please explain in words what each of the terms in the formula represents. Include a picture.

2) (6 pts each, 12 pts total) **Composition of 3D Transforms:** What is the sequence of transformations needed to achieve the operations given below. Also, include the corresponding inverse. You do not need to write out the 4x4 matrices. Instead, make use of the syntax:

Scale:  $S(s_x, s_y, s_z)$

Translation:  $T(t_x, t_y, t_z)$

Rotation:  $R_x(\Theta)$ ,  $R_y(\Theta)$ ,  $R_z(\Theta)$ .

a) A rotation of 35 degrees about the y direction, with fixed point (20,10,2).

The transforms:

The inverse:

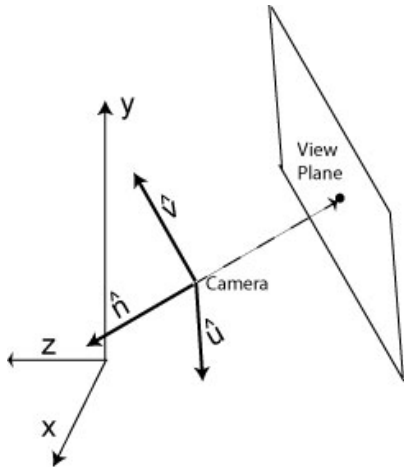
b) A scale by 2, with fixed point at (1,2,3), along the direction defined by the line from (0,0,0) to (1, 1, 0).

The transforms:

The inverse:

- 3) (20 pts total) **Camera Transforms:** Suppose you are given
- VPN = a vector that points in a direction *opposite* the way the camera looks
  - VUP = the up direction vector
  - Eye = location of camera in the World Coordinate System.

a) (6 pts) How does one calculate the normalized eye coordinate basis vectors:  $\hat{u}$ ,  $\hat{v}$ ,  $\hat{n}$  (see picture). Assume you are using a right handed coordinate system as shown.



$$\hat{u} =$$

$$\hat{v} =$$

$$\hat{n} =$$

b) (6 pts) How do you use  $\hat{u}$ ,  $\hat{v}$ ,  $\hat{n}$  and Eye to construct the View matrix, V, which transforms points from the World Coordinate System to the Camera Coordinate System?

c) (4 pts) When implementing fly-through navigation, how do you modify the View matrix so that the camera appears to turn to the right or left by some small angle  $\Theta$ ?

d) (4 pts) When implementing fly-through navigation, how do you modify the View matrix so that the camera moves forward by some small amount  $\alpha$ ?

- 4) (3 pts each, 9 pts total) **Model-View Transformations:** In class, we discussed how points are transformed from Object to Eye coordinates:

$$p_{eye} = V M p_{obj}$$

where  $V$  is the view matrix and  $M$  is the modeling matrix. Now suppose we have a standard x-axis rotation matrix  $R_x(\Theta)$  for some angle  $\Theta$ . Describe the effect on either the geometry and/or the camera's view when  $R_x$  is inserted as follows:

a)  $p_{eye} = V M R_x p_{obj}$

b)  $p_{eye} = V R_x M p_{obj}$

c)  $p_{eye} = R_x V M p_{obj}$

- 5) (8 pts) **Shading Algorithms:** What is the difference between Gouraud and Phong Shading? Which one is preferred and why?

6) (14 pts total) **Z-Buffer:**

a) (6 pts) What is a z-buffer? Why is it used? How does it work?

b) (4 pts) Why can it cause problems when implementing transparency?

c) (4 pts) How might the values in a z buffer be used to identify the edge boundaries between objects and the background?

7) (17 pts) **Shaders:**

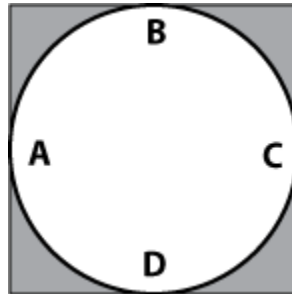
a) (9 pts) What is the difference between a vertex and fragment shader? Include as part of your answer: Which shader is executed first? What is a fragment? What are the minimal inputs/outputs of each shader?

b) (4 pts) What is the difference between an attribute variable and a uniform variable?

c) (4 pts) Vertex object coordinates are sent to the shader and transformed to eye coordinates using the modelview matrix. However, the position of a light is sent to the shader in eye coordinates. Why is it necessary to treat the light differently than the vertex coordinates?

8) (3 pts each, 12 pts total) **Texture Coordinates:** Suppose you want to use the image below as a texture on a square (e.g. a quad modeled as 2 triangles, although this doesn't matter for this problem).

image to be used  
as a texture object:



For each of the following possible texture coordinates, draw how the texture will look on the square. Assume that the texture wrap parameter is set to repeat in both dimensions.

<p>(0,1)                      (1,1)</p> <p>(0,0)                      (1,0)</p>	<p>(0,1)                      (2,1)</p> <p>(0,0)                      (2,0)</p>	<p>texture coordinates</p>
<p>(.5,.5)                      (.5,0)</p> <p>(0,.5)                      (0,0)</p>	<p>(0,1)                      (0,0)</p> <p>(1,1)                      (1,0)</p>	