NAME ____________________

# Final Exam
# Computer Graphics 462, Fall 2001

**Test-Taking Strategies:** read through all questions before you start writing, don't spend too long on a question if you don't know the answer, don't leave any question blank (give us a chance to give you partial credit!), and if you think a question is unclear without making some assumption, write down the assumption. And don't forget to put your name on the exam.
There are 10 questions, each worth 10 points.

**1. Rendering:**
a. Outline the key ideas behind the radiosity equation for rendering. Radiosity is most useful for what kinds of scenes?

b. How many levels of recursion will you need in a ray tracer to show the reflection on a sphere of a texture that is mapped onto one of the walls of the room? How many levels if a transparent sphere lies between the wall and the reflective sphere? Justify your answers.

c. What are the differences between flat shading, gouraud shading, and phong shading of polygons?

**2. Shadows and Hidden Surface Removal**

a. How can the first stage of a ray tracer be used for visible surface determination? Is this algorithm an image precision algorithm or an object precision algorithm?

**3. Transformations**

a. You are given a coordinate system that has been transformed with respect to the world coordinate system. The transformation matrix is

$$M = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0.8 & -0.6 & 1 \\ 0 & 0.6 & 0.8 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the coordinates of the point (1 1 1) in the transformed coordinate system?

b. Rotation transformations are not commutative. Demonstrate this by computing 1) the transformation matrix for a rotation by 90° about $x$ followed by a rotation by 90° about $y$ and 2) the transformation matrix for a 90° rotation about $y$ followed by a 90° rotation about $x$. 3) Apply the two composite transformation matrices to the point (1 1 1) to demonstrate that rotations are not commutative.

The formulas for rotation about $x$ and rotation about $y$ are

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**4. Modeling:**

a. Define the terms in the acronym NURBS.

b. Given the initiator and generator given below, draw the model that results from one and from two replacements:

c. ??

**5. Hidden Surface Removal:**
a. Write pseudo-code implementing a z-buffer algorithm for hidden surface removal.

b. What are the costs of this approach to hidden surface removal?

**6. Projections:**
a. List the sequence of operations (and purpose of each) in the transformation from world coordinates to screen coordinates using a parallel projection. You don't need to give the matrices, just the operations (e.g. shear) and a brief description of why they are needed (e.g. to align DOP with $z$ axis).

b. What is the difference between parallel and perspective projections? Describe an application where each type of projection would be preferable.

a. Ray tracing of an object requires calculating the intersection of that object with each ray. The parametric equations for a line between $(x_0, y_0, z_0)$ and $(x_1, y_1, z_1)$ are $x(t) = x_0 + (x_1 - x_0)t$,
$y(t) = y_0 + (y_1 - y_0)t$, and
$z(t) = z_0 + (z_1 - z_0)t$ for $t$ between 0 and 1.
Calculate the intersection of a line between (1, 2, 3) and (9, 12, 13) with the $x = 5$ plane.
b. One trick for speeding up shadow calculation for ray tracing scenes with many objects is to have a cube centered around the light source into which calculations are placed (this is called a light buffer). Explain the details of how this might be implemented. Why is it a savings?

c. If you could build your own virtual reality system, what would it look like? Describe the hardware, the types of user interaction and the domain. Is your VR system technically feasible with today's hardware?

**2. Transformations**

a. In Silicon Graphics's GL language the camera position can be specified with *lookat (fx, fy, fz, ax, ay, az);* where *fx, fy, fz* are the position of the camera in world coordinates and *ax, ay, az* are the location that the camera is looking at. If you had a view of a unit square with its center at $(0, 0, 0)$ and parameters of $fx = 5$, $fy = 5$, $fz = 0.5$, and $ax = 0$, $ay = 0$, and $az = 0.5$ what would the final image look like? Show the axes for the world coordinate system and a bird's eye view of the ground plane to explain how you arrived at the three-dimensional view.

This specification does not allow the user complete control over the camera position and orientation. What assumptions are inherent in the lookat command?

**3. Shadows**

a. Explain why visible surface determination and shadow generation algorithms address a similar problem. Why is resolution more of a problem for the shadow z buffer than for the hidden surface z buffer?

**4. Illumination and Interpolation**

a. Give the Phong illumination model including ambient, diffuse and specular components. Draw a picture to show the direction of the vectors in the equation.

b. What effects are lost in Phong illumination by putting the light source and the viewer infinitely far away from the scene?

**6. Ray Tracing I**
a. What factors make ray-tracing computationally expensive?

b. Explain the algorithm for two pass ray tracing.

**7. Ray Tracing II**

a. Name two effects that are missed by Phong illumination and interpolation that are captured by ray-tracing. Name one effect that is missed by both rendering techniques.

b. How many levels of recursion will you need in a ray tracer to show the reflection on a sphere of a texture that is mapped onto one of the walls of the room? How many levels if a transparent sphere lies between the wall and the reflective sphere? Justify your answers with a picture showing the rays.

**8. Miscellaneous:**

a. Outline the key ideas behind the radiosity equation for rendering. Radiosity is most useful for what kinds of scenes?

b. Why does stochastic ray tracing reduce aliasing in the image?

c. If you had to make a commercial tomorrow that included a running human, how would you animation the motion of the human?