Math:

1. [9] Suppose you have a triangle rasterizer that fails if the three vertices of a triangle are colinear. Devise a test for colinearity of the three, three-dimensional points. You may describe your test using simple statements and mathematical operators.

2. [9] Write an equation for a plane in three dimensions given three points specified by a user: $\vec{P}_0, \vec{P}_1, \vec{P}_2$. Your equation may include mathematical operators.

Lighting:

3. [3] Which of the following is the best example of a Lambertian surface?
   
a) Mirror  
b) Chalk  
c) Apple  
d) Glass
4. [6] Johnny has written some code to draw the following unit cylinder:

![Unit Cylinder Diagram]

His drawing code places vertices along the upper and lower rims of the cylinder, connecting them together with triangles that stretch the entire length of the cylinder. He draws the cylinder using a shiny material. Given the position of the light, he expects to see a specular highlight in the middle of the cylinder.

a) When Johnny executes his OpenGL code (using Gouraud shading) no highlight appears in the middle of the cylinder. However, when he renders it with RenderMan (using Phong shading) the highlight appears. Explain why this is.

b) How could the OpenGL code be altered (still using Gouraud shading) so that the highlight would appear?
5. [6] The radiosity of a patch ‘i’ is: \[ B_i = \varepsilon_i + \rho_i \sum_{1 \leq j \leq n} B_j F_{ji} \]. Define the terms.

\( \varepsilon_i \)

\( \rho_i \)

\( B_j \)

\( F_{ji} \)

**Aliasing:**

6. [6] The nVIDIA GeForce2 from last year used supersampling to accomplish full screen antialiasing whereas this year’s GeForce3 uses multisampling. Briefly describe the two techniques:

**Supersampling:**

**Multisampling:**

7. [3] We use the _________________ transform to convert an image from the spatial domain to the frequency domain.

8. [6] What is the Nyquist rate and how does it affect how we construct images?
Texture Maps:

9. [9] Given two three-dimensional points that define a line, $L_0$ and $L_1$, build a parametric equation for the line and solve for $t$, the parameter value of the intersection point between the line and a plane, $Ax + By + Cz + D = 0$.

$$t =$$

10. [6] (a) Why does a texture that is 256 texels by 256 texels alias (look bad) when it is applied to a polygon that fills only 1,000 pixels on the screen?

10(b) Why would a MIP map improve the quality in this situation?

11. [3] What attribute of a polygon is affected by a bump map?
Visibility:

12. [6] What are three reasons for why a polygon would not need to be rendered?

13. [6] Consider the following scene:

a) In what order must the triangles in this image be drawn if the Painter’s algorithm is used?

b) Suppose we are using Z buffering. What drawing order would redraw the fewest number of pixels.
14. [6] Draw the lines that will partition the objects in this 2-D world according to the BSP algorithm. Label your lines. On the right, draw the resulting binary tree (with nodes and leaves labeled).

15. [6] Why is Warnock's algorithm inefficient for scenes with many small polygons?

Rotations:

16. [6] What is gimbal lock? Give an example of rotation that causes gimbal lock (when the order or rotations is defined as \( x \rightarrow y \rightarrow z \)).
17. [9] You are given a unit square centered on the x-z plane. You want this unit square to represent a clipping plane defined by $Ax + By + Cz + D = 0$. The unit square can be drawn in the correct location (aligned with the clipping plane and centered on the line that is normal to the clipping plane and passes through the origin) using one glRotate(), one glTranslate(), and one glBegin()/glEnd(). Write pseudocode that renders and transforms the square (make sure your operations are in the correct order). Your answer may include mathematical operators.

18. [9] The following image depicts the process of rotating a vector, $r$, by $\phi$ degrees about another vector, $\hat{n}$. Define the vector labeled with a $V$ in the image in terms of $r$ and $\hat{n}$.

$$V =$$
Splines:

19. [12] Given the definition of a general cubic spline in one dimension:

\[ f(t) = at^3 + bt^2 + ct + d; \quad 0 \leq t \leq 1, \]

and the user specifications for a Hermite spline: \( f(0), f(1), f'(0), \) and \( f'(1), \)

a) fill in the [four by four] matrix that is used to solve for the four unknowns in the following equation:

\[
\begin{bmatrix}
  f(0) \\
  f(1) \\
  f'(0) \\
  f'(1)
\end{bmatrix}
= \begin{bmatrix}
a \\
b \\
c \\
d
\end{bmatrix}
\]

b) write the definition of the cubic spline using the Hermite blending functions:

\[ f(t) = \]

20. [3] We wish to use a recursive spline subdivision algorithm to render this Bezier spline as a series of straight lines. This spline will be subdivided into two halves and recursively subdivided. Which four points will be used as arguments to specify the left half of this spline for the recursive rendering algorithm?
21. [6] Indicate which of the following are not Bezier curves (there may be none or more than one) and write why each cannot be a Bezier curve.

\[ P_0 \quad P_1 \quad P_2 \quad P_3 \]

\[ P_0 \quad P_1 \quad P_2 \quad P_3 \]

\[ P_0 \quad P_1 \quad P_2 \quad P_3 \]

\[ P_0 \quad P_1 \quad P_2 \quad P_3 \]

**Animation and Movies:**

22. [6] You are working on the Disney production of 101 Dalmations. In your scene, all the dogs will be computer generated. What animation technique(s) will you use to generate the desired motion, and why? Please provide two reasons.

Reason 1)

Reason 2)
23. [6] You’ve been hired to work on Terminator 3 and you are working on a scene that combines live-action film with a computer graphic character. In the live-action scene, the camera pans to follow (only allowed to rotate on a tripod) an actor and your rendered computer graphic character must stand in the background. List two pieces of information you would like to obtain from the live-action filming crew in order to simplify the post-process computer graphics work? How will you obtain (record) the information you need?

Item 1)

Item 2)

Extra Credit [6]: We read a web posting and discussed in class how cubic environment maps are done in hardware. Briefly describe how the hardware solution to environment mapping is different from the OpenGL version we developed in the last assignment.