Practice Final
CMSC 427

General Guidelines: The final will focus on topics that have been discussed in class. However, there will be no questions about OpenGL or programming problems. You will be expected to know some basic equations, but nothing very obscure. I'll try to indicate below the key equations you should know.

The final will be closed book, with no notes or calculators. This means that problems should not require the use of a calculator; if you feel you need one when you are doing a problem this probably means there is an easier way to do the problem.

The final will be comprehensive.

Topics and things to know about them:

1. Geometry (needed for problems on the next two topics)
    a. How to take an inner product, and what this does.
    b. How to take a cross product, and what this does.
    c. What a unit vector is, and how to find the magnitude of a vector.
    d. How to write the equations for a 3D line, or for a plane in 3D, or for a sphere.
2. Transformations
    a. How to write rotation about the *x, y,* or *z* axis as a matrix.
    b. How to write translation as a matrix.
    c. How to combine these. For example, rotate about a point not at the origin.
    d. How to relate the rows of a rotation matrix to the axes of a new coordinate system
        This includes writing the coordinates of points relative to a new coordinate system.
        Given a position, lookat vector that indicates the direction a viewer is looking, and upvector, indicating which way is up, you should be able to write a matrix that transforms the world so that the viewer is at the origin, with the lookat direction in the *z* direction, and the upvector in the *y* direction.
3. Projection
    a. Perspective projection
        i. How to find the image coordinates of a 3D point
        ii. Some consequences of that, such as properties of vanishing points, knowing when a line in 3D does not project to a line in the image, …..
    b. Orthographic projection
        i. How to find the image coordinates of a 3D point.
        ii. Consequences of orthographic projection, such as that it preserves parallelism and ratios of areas. If a 3D point is inside a 3D triangle, the 2D orthographic projection of the point will be inside the 2D orthographic projection of the triangle.
4. Intersections and Relations between lines and planes and points.

a. How to use the equation of a line or plane to tell which side of the line/plane a point is on.
   Example: Given a plane and two points, are the points separated by the plane, or on the same side of it?
b. How to find the intersections of spheres, axial rectanguloids, triangles.
c. How to intersect a line with a plane, triangle, or sphere
d. How to tell if a point is inside a polygon.

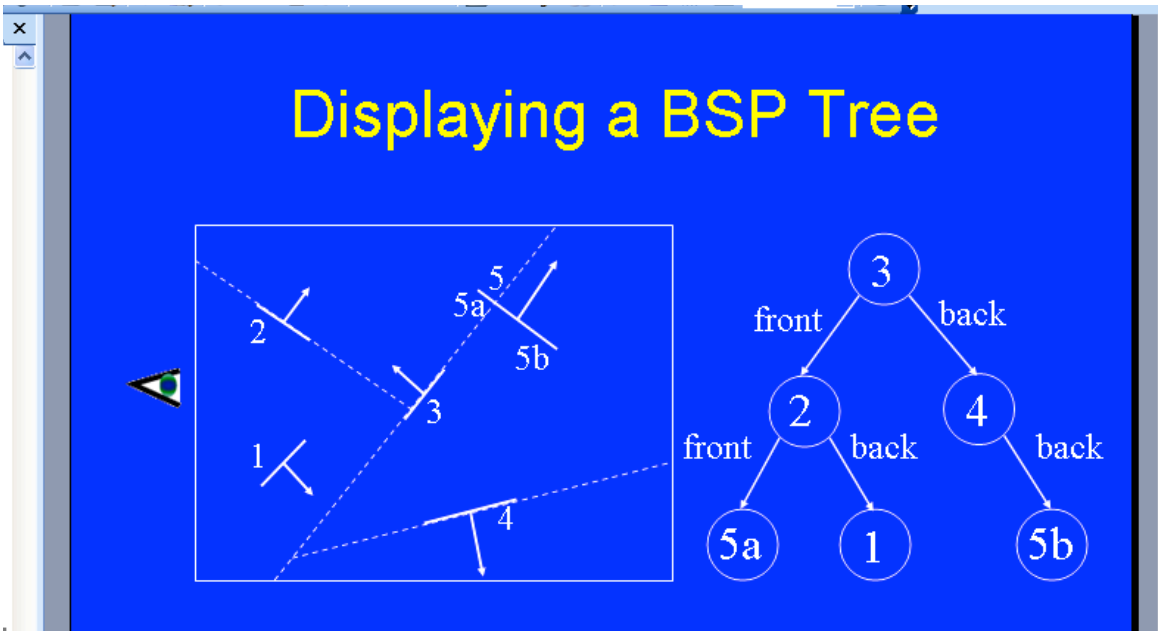
5. Sampling and Aliasing
   a. Creating an image by sampling
   b. Aliasing – what it means, what causes it.
      i. Spatial
      ii. Temporal
   c. Anti-aliasing
      i. Smoothing
      ii. Supersampling and smoothing
6. Interpolation
   a. Linear, bilinear
7. Color and matting
   a. Superposition
   b. RGB representations of color. Why color is three-dimensional. Why some colors cannot be reproduced using RGB.
   c. What is alpha matting.
8. Visibility
   a. Culling – Basic idea
   b. Painter's algorithm
      i. For example, how to tell if a triangle 1 is in front of, or behind, the plane that contains triangle 2, and how knowing this might help you in the painter's algorithm.
   c. Z buffer.
      i. Given the vertices of a 3D triangle, you should be able to figure out where they appear in an image (perspective projection), which 2D points are inside the triangle, and what their depth is.
   d. BSP Trees – The basic idea of the algorithm. See the problem below.
9. Lighting – Optics and Reflectance
   a. Ambient – What does this mean?
   b. Lambertian -- Know the equation for Lambertian reflectance.
   c. Specular –
      i. Know the equation for Phong reflectance.
      ii. Know how to figure out the direction of mirror reflection.
10. Ray Tracing
    a. Shadows
    b. Finding rays from focal point to pixels.
    c. Given the intersection point between a ray and an object, know in which directions you need to cast additional rays.
11. Filtering
    a. How correlation and convolution work.

b. How to filter an image to reduce noise, using an averaging or Gaussian filter.
c. How to construct a discrete filter (such as a Gaussian) from a continuous function.
d. Sharpening filters.
e. Resizing images
f. Separating a 2D filter into two 1D filters

12. Shadows
    a. Z-buffer algorithm – just the basic idea.
    b. Casting a shadow with a directional or point source onto a plane.
13. Modeling
    a. Implicit shapes – Don't memorize which shapes have which equations (except you should know the equation for a sphere).
    b. Kinect – the basic idea of how it works.

Sample Problems (Note, some of these problems may be a bit more involved than ones I'd ask on a time-limited exam).

1. Suppose the camera's focal point is at (1,2,3), you are looking in the direction (1,1,1), and the focal length is 1. Give a ray from the focal point through the center of the image.
2. Suppose the camera has a focal point at (0,0,0), and you cast a ray through a pixel whose center is at (1/2, 0, 1). Does this ray intersect a sphere centered at (3, 2, 12) with a radius of 3?
3. Suppose the camera has a focal point at (0,0,0), and you cast a ray through a pixel whose center is at (1/2, 0, 1). This ray definitely intersects a sphere centered at (3, 2, 12) with a radius of 5. What are the coordinates and surface normal of the first point that the ray intersects?
4. Consider the BSP tree discussed in class:



a. Suppose we look at this scene from the right side, instead of the left. Use the BSP tree to determine the order in which the triangles should be rendered.
b. Construct a valid BSP tree starting with triangle 1 instead of triangle 3.

5. Suppose you want to model the reflectance properties of a blue car.
   a. What reflectance model might you choose, and what parameters might you use? Why? You might want to limit yourself to using the models we've discussed in class (eg., Phong, Lambertian).
   b. Suppose you have a flat piece of blue car hood. You shine a light on the car from an angle of pi/4. Then you take a few pictures of the car from different angles. How might you use these images to determine what parameters you should use to model its reflectance? How many images do you think you would need?
6. Create a simple scene with a viewer looking in a specific direction, one or two simple objects, and a point source of light. Then work out how to render or ray trace this scene. Figuring out what kind of scene will create a reasonable set of problems will also be helpful to your studying. This is a Meta-problem. If you can do this, you have studied much of what is needed.
   a. Transformations
      i. Create a matrix that will transform the world so that the viewer is at the origin and the viewing direction is in the z direction.
      ii. Create a matrix that will transform the world so that the viewer is at the origin and the viewing direction is in the z direction.
      iii. Apply the equations of perspective projection to determine the image location of the object.
   b. Ray Tracing.
      i. Find the equation for the image plane.
      ii. Determine a ray that goes from the viewer to a vertex in the scene, and figure out where it intersects the image plane.
      iii. Determine a ray that goes through the center of the image, and figure out which object in the scene it first intersects.
      iv. Suppose a ray strikes a surface that is a bit shiny (that is, it reflects light with some Phong and some Lambertian reflectance), but it is not at all transparent. If we are allowing for multiple bounces and shadows, what rays will we cast from the point where the original rays hits the surface. Give the direction of each ray.
   c. Lighting
      i. Find a ray from the viewer that intersects an object in the scene. Suppose the object is Lambertian and white. Determine the intensity it will produce in the image.
      ii. Now do the same, assuming the object has Phong reflectance.
7. Filtering
   a. Show how you can divide a 3x3 box filter into two, 1D box filters.
   b. Suppose I want to smooth an image using a filter like an upside down parabola. That is, I'll base the filter on the equation $y = 16 - x^2$, but

only using positive values of the equation. How would I construct this filter?