A1 Computer Graphics (25 points)

Shadow maps are data structures that can be used to quickly render hard shadows in a local illumination pipeline. The idea is based on computing a z-buffer from each light's point of view.

a. [5 points] Explain the standard purpose of z-buffering, both in terms of application and algorithmic detail.

b. [12.5 points] How might a regular z-buffer be modified and used, step-by-step, to render shadows as hinted above?

c. [7.5 points] Certainly shadow mapping can be faster than ray tracing, but what are some of its negative aspects? Explain.

A2 Computer Graphics (25 points)

Suppose you want to rasterize a Bezier curve p(t) defined by the 2-D control points p1, p2, and p3 over the interval 0 <= t <= 1.

a. [15 points] Taking inspiration from the midpoint/Bresenham's algorithm for line rasterization, describe in detail an incremental (i.e., compute as little as possible at each step) algorithm for drawing a piecewise linear approximation of the curve at delta t intervals. Assume that you already have an efficient line-drawing function as a subroutine, draw_line(x1, y1, x2, y2).

b. [10 points] The algorithm above will produce aliased curves. What does this mean and what properties would an anti-aliased version of the curve have? Assuming a pen thickness of 1 pixel, discuss how the algorithm might be modified to achieve anti-aliasing. Can you see any potential issues with your approach?

A3 Computer Graphics (25 points)

In this section, you will be asked to find matrices which will transform a point p into another point p'. Assume p and p' are given in homogeneous coordinates. Your answers do not need to be simplified, i.e., you may leave your answers as a product of several matrices instead of simplifying them to one matrix.

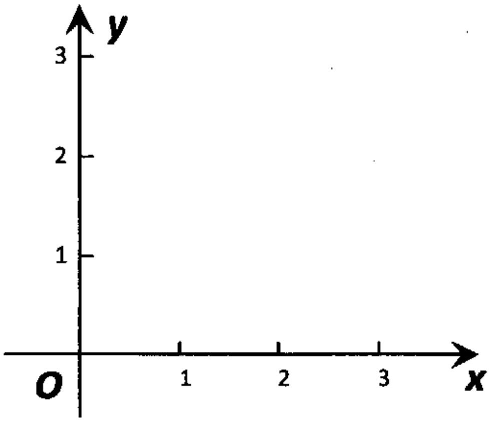
a. [5 points] Assume R and S are both two dimensional transformations. Let R be the rotation matrix and let S be a uniform scaling matrix that scales the x and y coordinates the same $s_x = s_y$). Is the product of R and S commutative, i.e., is RS = SR?

b. [5 points] The equation $[3, 5, 6]^T + [0, 1, 0]^T t$ defines a line in parametric form in t. What is the 4x4 matrix that will rotate a point about the line by θ degree counterclockwise? Assume the right hand rule when rotating about the positive y-axis.

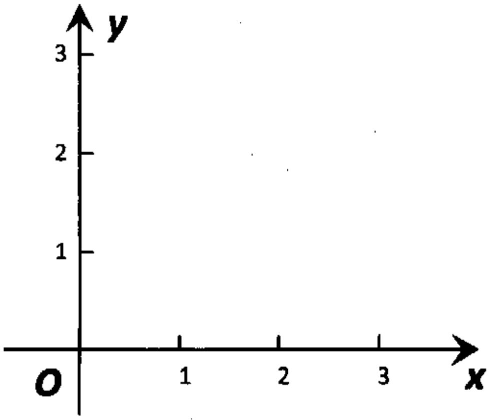
c. [5 points] Assume the center of an object is located at the point (2, 3, 4). What is the 4x4 matrix that will scale down an object by 30% in towards the object's center?

d. [10 points] Consider the following two sections of OpenGL code that both draw a line segment. With your pen, draw the line on the figure at right to show where the line would appear in each case.

```
gl.glLoadIdentity();
gl.glRotatef(45, 0, 0, 1);
gl.glTranslatef(1, 0, 0);
gl.glBegin(GL.GL_LINES);
   gl.glVertex3f(0, 0, 0);
  gl.glVertex3f(1,0, 0);
gl.glEnd();
```



```
gl.glLoadIdentity();
gl.glTranslatef(1, 0, 0);
gl.glRotatef(45, 0, 0, 1);
gl.glBegin(GL.GL_LINES);
   gl.glVertex3f(0, 0, 0);
  gl.glVertex3f(1,0, 0);
gl.glEnd();
```



A4 Computer Graphics (25 points)

a. [5 points] We want to render a scene of N objects and K point lights at an image resolution of $\sqrt{M} \times \sqrt{M}$ (thus there are M pixels). Without any acceleration structure (which means you have to intersect each ray with every object to find the closest intersection point), what is the complexity of ray tracing the whole image?

b. [5 points] How do you think the complexity will change if you have a good ray tracing acceleration structure, such as a kd-tree.

c. [15 points] A plane passes through a point O and is perpendicular to a normalized vector N.

1. [8 points] Write down the plane equation (Hint: as an implicit function).

2. [7 points] Write down a formula to compute the distance from an arbitrary point Q in space to the plane?