**NAME:**

Login name:

---

# Computer Science 426 Midterm
# 7PM-9PM

---

This test is 8 questions, of equal weight. Do all of your work on these pages (use the back for scratch space), giving the answer in the space provided. This is a closed-book exam -- you may use one-page of notes with writing on both sides during the exam. **Put your name on every page, and write out and sign the Honor Code pledge before turning in the test.**

***''I pledge my honor that I have not violated the Honor Code during this examination.''***

| Question | Score |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| Total: | |

**NAME:**

## Q1: Color Models

a) Draw the RGB cube. Label the axes and vertices.
Draw and label the region representing gray values.

b) Draw the HSV hexcone. Label the axes.
Draw and label the region representing gray values.

c) Can every color perceptible to the human eye be displayed as a combination of three primary colors visible to the human eye? Give evidence to support your answer using the CIE Chromaticity diagram.

**NAME:**

## Q2: Scan Conversion

a) What is the odd-parity rule for testing if a point is inside a polygon?

b) What is the nonzero winding number rule?

c) Give an example polygon for which the two rules produce different results.

d) Which rule does the sweep-line algorithm implement?

**NAME:**

## Q3: Modeling Transformations

For each of the following cases, write a 4x4 matrix which applies the specified transformation to any 3D point (x,y,z,1). Assume that points to be transformed are represented as a column vectors and they are pre-multiplied by the matrix. If it is impossible to define a matrix for the given transformation, say so and explain why.

a) Transform ( x, y, z, 1 ) to ( 4x+2, -3y - 2, z/2, 1):

b) Transform ( x, y, z, 1 ) to ( x, y, 1, z):

c) Transform ( x, y, z, 1 ) to ( xy, yz, xz, 1):

d) Scale (x, y, z, 1) by a factor S around an arbitrary "center of scale" point C = (cx, cy, cz):

e) Mirror (x, y, z, 1) over the plane represented by x - y = 0:

**NAME:**

## Q4: Viewing Transformations

a) Circle each of the following statements that is true for PARALLEL projections:

Lengths vary with distance to the eye

Angles are not preserved, in general

Parallel lines remain parallel

All parallel projections are linear

b) Draw a picture depicting the perspective projection of a point at (x, y, z) onto the view plane in a right-handed camera coordinate system. Be sure to label the camera coordinate system origin, the camera coordinate system axes (X, Y, and Z), the original point (x, y, z), the view plane (which is D units from the origin), and the projected point ($x_s$, $y_s$, $z_s$).

c) Write the equations for the projected point ($x_s$, $y_s$, $z_s$) on the view plane in terms of (x, y, z) and the distance D from the origin to the view plane.

d) Write the equations from part (d) with a 4x4 matrix M. Assume the point (x, y, z, 1) in homogeneous coordinates is a column vector that will be premultiplied by M to form the projected point ($x_s$, $y_s$, $z_s$, $w_s$) in homogeneous coordinates.

**NAME:**

## Q5: Bezier Curves

a) How many control points are required to specify a Bezier curve of degree d?

b) Circle each of the following statements that are true for a cubic Bezier tensor product surface:

Interpolates its four corner control points

Interpolates the centroid of its control points

Lies within the convex hull of its control points

Has at most one point with positive curvature

c) Draw the cubic Bezier blending functions -- i.e., the Bernstein polynomials (B1, B2, B3, B4).

d) What property of the Bezier blending functions insures that a cubic Bezier curve interpolates the first control point (V0)?

**NAME:**

## Q6: Hidden Surface Removal

a) Please briefly describe how the z-buffer hidden surface removal algorithm works. Please keep your answer very short -- a few concise sentences should be sufficient.

b) List at least two important advantages and two significant disadvantages of the Z-buffer algorithm as compared to other hidden surface removal algorithms.

c) If writing shaded pixels into the frame buffer is hypothetically the ***only*** performance bottleneck in the rendering pipeline, rank the following hidden surface removal algorithms from fastest to slowest. Write “1” next to the fastest, “2” next to the second fastest, amd “3” next to the slowest.

Painter’s algorithm (front-to-back depth-sort)

Z-buffer

Casting a ray through each pixel

**NAME:**

## Q7: Radiosity

a) Write the radiosity equation.  Define each of the terms.

b) Circle each of the following assumptions that must be true for the radiosity equation to be a good approximation to the rendering equation:

Emitting surfaces do not also reflect light

The radiosity is the same at all points on a patch element

There are no patches that block light transfers between any two other patches

All surfaces are purely diffuse

c) Circle each of the following lighting effects modeled by solving the radiosity equation:

Soft shadows

Direct illumination from area light sources

Indirect illumination due to inter-object specular reflections

Indirect illumination due to inter-object diffuse reflections

**NAME:**

## Q8: Polygonal Meshes

a) What is a triangle strip?
Draw a picture, label the vertices, and draw a suitable data structure.

b) Explain why it is more efficient to perform lighting calculations for a triangle strip than a list of independent triangles when using Gouraud shading?

c) What is the computational complexity (e.g., O(logn)) of splitting an edge (inserting a new vertex between to given vertices V1 and V2) in each of the following mesh representations with N triangles. Justify your answers.

- List of independent triangles (like the .ray representation):

- Triangle strip/fan:

- Vertex table and face table with references to vertice

- Winged-edge: