Milwaukee School of Engineering

Electrical Engineering and Computer Science Department

CS421 – Midterm

Thursday 22 January 2003

Allowed Materials: 8½" × 11" Note Sheet, 1 Side; Calculator

Name: _____

Problem 1:   (40 Points) _____

Problem 2:   (10 Points) _____

Problem 2:   (25 Points) _____

Problem 3:   (25 Points) _____

Total: (100 Points) _____

Use this space for additional work, if needed.

Problem 1 – Short Answer / True/False (40 Points)

A. (6 points) Give 3 examples of OpenGL state variables.

B. (4 points) OpenGL uses function name suffixes such as "3fv" to indicate the types of arguments that a function takes (e.g., 3 pointers to float arrays). What is a reason for doing this instead of using overloads?

C. (6 points) Give 2 examples of a rigid transformation and 1 example of a non-rigid transform.

D. (6 points) Information about polygon facing is provided to OpenGL by both `glFrontFace(…)` and `glNormal3*(…)`. Briefly describe one use of each type of facing data.

E. (5 points) Give a brief definition of double buffering and explain why it is often used in animations.

F. (4 points) Depth buffering and backface culling are two different methods for determining whether part of an image is visible. Assume a model in which backface information has been properly set up and in which objects are only viewed from the outside. What is the advantage to enabling culling if the depth buffer is already being used?

G. (2 points) True or false: "The last transform specified for the MODELVIEW matrix is the first one logically applied to each vertex."

H. (5 points) Explain "glPushMatrix" and "glPopMatrix". Be specific about how they affect the current transformation and the stack.

I. (2 points) True or false: "As long as I avoid shearing transformations and specify normals to have unit length, it is not necessary to enable one of the OpenGL renormalization features to ensure proper surface normal calculation."

Problem 2 – Viewing Volumes (10 Points)

Sketch a viewing volume selected by gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar) and **label or describe** the effect of each argument on the volume.

Problem 3 – Animations with QGLWidget (25 Points)

An animation program is using Qt and OpenGL (using the default double-buffering feature). The constructor of the object derived from QGLWidget creates a QTimer and connects its timeout event to paintGL. That widget then start()s the timer in its polish() function. Note that the start() function is called with arguments that cause the timer to fire every 1/25th of a second.

Each time paintGL() is called, it increments a frame number counter and then renders the current frame. **What is a problem** that this will cause on very slow graphics hardware? **How** might we improve video quality on slow graphics hardware?

Note: There are many right answers to this question; you only need to provide one. It is not necessary to use your OpSys knowledge to answer this question, but depending upon the approach you take, you might find it useful to know that the application runs in a single thread and that signals cannot preempt this thread, although there is a function that long-running functions can call to allow waiting signals to fire.

Problem 4 – Solid Rendering and Normals (25 Points)

An icosahedron is a 20-sided regular solid with triangular faces. Four of the vertices of a particular icosahedron centered at the origin are...

	x	y	z
A	1.618	0	1
B	0	1	1.618
C	0	-1	1.618
D	-1.618	0	1

Two of the faces are given by triangles ABC and BCD. ABC is specified such that the vertices are in a counterclockwise direction when viewing the solid from the outside. This is not true for BCD.

(Note: 1.618 is an approximation of $\dfrac{1+\sqrt{5}}{2}$.)

Write the OpenGL calls that will draw the first 2 faces and properly **establish** the outward surface normal for each face. You may assume that normalization is enabled; your normal vectors must point in the right direction, but do not need to be of unit length.

Extra Credit (5 Points): **Add** culling support to your geometry.