COMP 341 Computer Graphics
Midterm Examination, 2001 Spring Semester

Name: ______________________ E-mail: ______________________
Student No. ______________________ Tutorial Session: ______________________

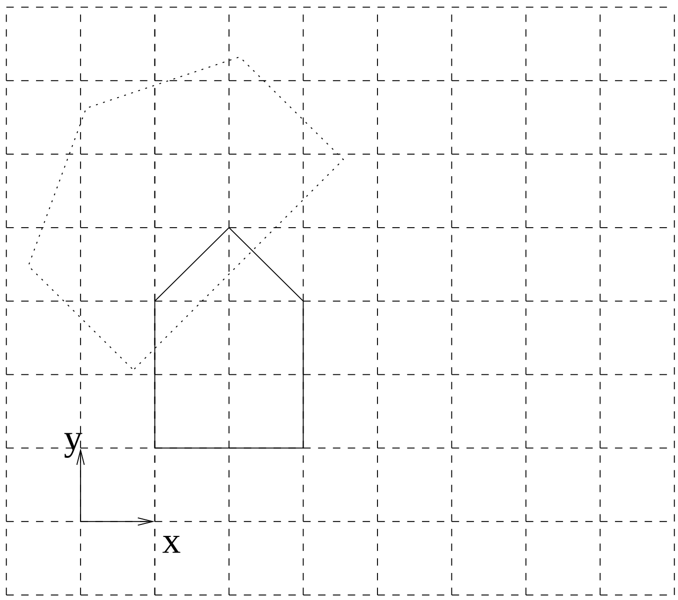
1	
2	
3	
Total	

You have one and a half hour to solve 3 problems for a total of 100 marks. Answer all problems in the space provided. You may also use the back of the test papers. Good luck!

Problem 1. (30 points)

a) (10 points)
Suppose a procedure `drawHouse()` draws the shape shown in figure below. Sketch (in the same figure) the shape drawn by the following OpenGL code:

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef(45.0, 0.0, 0.0, 1.0);
glScalef(2.0, 1.0, 1.0);
drawHouse();
```



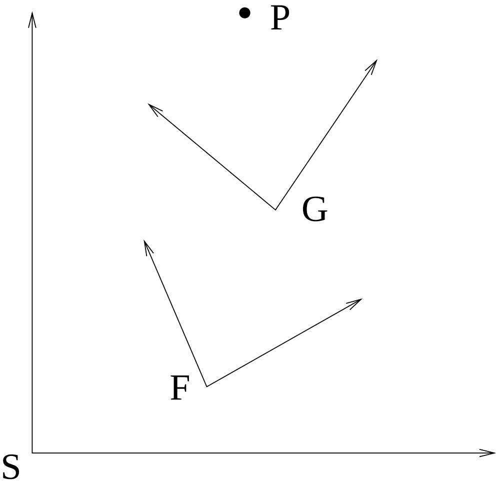
b) (20 points)
Given three coordinate frames: the standard frame S, frame F, and frame G, where F is defined with respect to S, and G is defined with respect to F; that is,

the elements of frame S are $(S.\vec{e}_0, S.\vec{e}_1, S.\mathcal{O})$;

the elements of frame F are $(F.\vec{e}_0[S], F.\vec{e}_1[S], F.\mathcal{O}[S])$;

the elements of frame G are $(G.\vec{e}_0[F], G.\vec{e}_1[F], G.\mathcal{O}[F])$.

If the representation of a point P with respect to frame G is $(a, b, 1)^T$, what is its representation with respect to frame S?

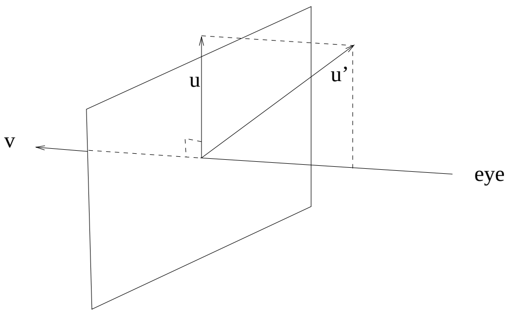


Answer:

$$
\begin{aligned}
P[F] &= G.\mathcal{O}[F] + a \cdot G.e_0[F] + a \cdot G.e_1[F] \\
&= (G.e_0[F] | G.e_0[F] | G.\mathcal{O}[F]) P[G] \\
P[S] &= (F.e_0[S] | F.e_1[S] | F.\mathcal{O}[S]) P[F] \\
&= (F.e_0[S] | F.e_1[S] | F.\mathcal{O}[S])(G.e_0[F] | G.e_0[F] | G.\mathcal{O}[F]) P[G]
\end{aligned}
$$

Problem 2. (35 points)

a) (15 points)
Given the viewing direction $\vec{v}$ and the upward vector $\vec{u'}$. Assume both vectors are of unit length and that the projection plane is perpendicular to the view vector. One way to determine the y-axis of a viewing frame is by projecting $\vec{u'}$ onto the projection plane. Give an expression for the projected vector $\vec{u}$.



Answer:
Let $\vec{u''}$ denote the projection of $\vec{u'}$ onto $-\vec{v}$. The magnitude of $\vec{u''}$ is

$$|\vec{u''}| \cos(\text{ angle between } \vec{u'} \text{ and } -\vec{v}) = -\vec{u'} \cdot \vec{v}$$

Thus,

$$\vec{u''} = (-\vec{u'} \cdot \vec{v})(-\vec{v}) = (\vec{u'} \cdot \vec{v})\vec{v}.$$

Since $\vec{u} + \vec{u''} = \vec{u'}$, we get

$$\vec{u} = \vec{u'} - \vec{u''} = \vec{u'} - (\vec{u'} \cdot \vec{v})\vec{v}$$

b) (20 points)
In OpenGL, what are the homogeneneous coordinates of the eye after it is transformed by the perspective transformation matrix? (you may leave your answer in terms of, say, α, β, etc.) What happen if you perform perspective division to this point?

Answer:
Before applying the perspective transformation, the eye is at the origin of the eye coordinate frame, i.e., eye $= (0, 0, 0, 1)^T$. The perspective transformation matrix is

$$M_{persp} = \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \phi & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Thus,

$$M_{persp}(0, 0, 0, 1)^T = (0, 0, \beta, 0)^T$$

Performing perspective division to this point will give division-by-zero error.

Problem 3. (35 points)

a) (15 points)
Suppose the diffuse and specular coefficients of a nonblack object are both nonzero. Assume that there is only one light source and that we are using Phong illumination model. Is it possible for a point on this object to be the brightest specular point, but does not reflect any diffuse light? Explain your answer in terms of the vectors used in Phong model.

Answer:
By the definition of the light-reflection vector $\vec{r}$, $angle(\vec{l}, \vec{r}) \leq 180°$. For a point to be the brightest specular point, $\vec{v} = \vec{r}$, thus $angle(\vec{l}, \vec{v}) \leq 180°$. Claim that $angle(\vec{l}, \vec{n}) < 90°$, because otherwise the back of the polygon would be facing the viewpoint, which either means it is blocked by some front-facing polygons and hence cannot be a specular point, or it is visible and the surface normal has to be reversed.

b) (20 points)
Under what situation(s) would you get $\vec{h} \cdot \vec{n} < 0$ in OpenGL lighting model? Sketch a scenario.

Answer: When $\vec{h} \cdot \vec{n} < 0$, we have the light and the eye on opposite side of a polygon as shown below: