CMPS 160 – Midterm Exam #1

Name / Username (unix.ic)

CMPS 160 F10
Introduction to Computer Graphics
Nov 23, 2010

Midterm 2 Exam

You have the entire class period to complete this exam.

All pages are worth an equal amount.

Partial credit will be given for clear evidence of correct reasoning even if the final solution is incomplete.

No books
One page hand written notes
Simple calculators allowed
No computers
1. For each call to `drawObject` below, identify the matrix that transforms the modeling coordinates resulting from the `drawObject` call to the final device coordinates used when rasterizing. Report this matrix as a product of matrices mentioned in the code (ex: "DBCA") assuming the scene's vertices (represented as column vectors) will be placed to the right of your matrix for multiplication (this is the convention in OpenGL). Watch out, the code may do things that would be silly to put in a real program.

The function `glMultMatrix` is part of the OpenGL specification and works much like many of the other matrix manipulation functions. If $M_1$ was the state of the current matrix (as selected with `glMatrixMode`), then $M_1M_2$ is the state of the matrix after a call to `glMultMatrix` with $M_2$ as the argument.

```
GLMATRIXMODE(GL_PROJECTION);
GLLoadIdentity();
glMultMatrix(A);

drawObject();

\[ A \]

drawObject();

\[ AB \]

glMultMatrix(B);

drawObject();

\[ ABC \]

glMultMatrix(C);

glPushMatrix();

glMultMatrix(D);

drawObject();

glPopMatrix();

drawObject();

glMultMatrix(E);

glPopMatrix();

drawObject();

glLoadIdentity();

GLMATRIXMODE(GL_PROJECTION);
GLLoadIdentity();
drawObject();

\[ \text{Identity} \]

glMultMatrix(F);

glPushMatrix();

glMultMatrix(G);

glPushMatrix();

drawObject();

glPopMatrix();

drawObject();

\[ FG \]
```

Any other single answer incorrect (-1)
2. Color Spaces

a) Sketch the HSV and RGB color spaces with labels for each of the axes and indicate which location each of the colors black, white, gray, red, cyan, and yellow occupy in each color space. (In HSV Red is at 0 degrees and Green is at 120 degrees)

b) In the RGB color model, starting with a pure green color, describe the effect that moving towards white, black, etc... has on R, G, and B. Define "moving towards" as moving in the RGB cube in the direction of the new color. Use the terms constant, increase, or decrease to describe the change.

<table>
<thead>
<tr>
<th>Move towards white</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase</td>
<td></td>
<td></td>
<td>Increase</td>
</tr>
<tr>
<td>Constant</td>
<td></td>
<td></td>
<td>Constant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Move towards black</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>Decrease</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Move towards gray</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase</td>
<td></td>
<td>Decrease</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Move towards yellow</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase</td>
<td></td>
<td></td>
<td>Increase</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Move towards cyan</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td></td>
<td>Increase</td>
</tr>
</tbody>
</table>

+4  - 14 to 15 correct (-1)
+3  - 13 to 13 correct (-2)
+2  - 10 to 10 correct (-3)
+1  - 9 to 9 correct (-4)
3. Frame buffer / Zbuffer
Suppose we have a graphics system that rasterizes polygons by turning on pixels that are strictly inside the bounds of the polygon. This is an orthographic rendering looking down the positive Z-axis. For example:

a. Show the state of the z-buffer (with numbers written at grid intersections) once this polygon has been rendered.

```glBegin()
glVertex(0,0,1)
glVertex(5,5,1)
glVertex(5,0,1)
glEnd()```

b. Without clearing the z-buffer, what is the state after adding this polygon.

```glBegin()
glVertex(0,0,4)
glVertex(0,5,4)
glVertex(5,5,4)
glEnd()```

c. Without clearing the z-buffer, what is the state after adding this polygon (not a square).

```glBegin()
glVertex(1,1,2)
glVertex(4,1,2)
glVertex(4,4,2)
glVertex(0,4,2)
glEnd()```

(-2) If you write on the polygon borders but get everything else correct.

(-2) Depth buffer labels are not depth value, but otherwise correct.

(6) Points for just the correct polygons, no labels.
4. Lighting models

a) Write the OpenGL lighting equation using these symbols $k_a$, $k_d$, $k_s$, $L$, $N$, $V$, $R$, $n_e$.

$$k_a + k_d \left( N \cdot L \right) + k_s \left( R \cdot V \right)^n$$

b) Draw a picture of lighting on a surface at point x. Labeling the light and eyepoint. Also label the vectors for $L$, $N$, $V$, $R$.

![Diagram of lighting](image)

c) Suppose the surface point $x$ is at (1,2,3), the normal is (0,0,1), the light is at (4,5,3), and the camera is at (1,2,6). What are the normalized vector values of $L$, $N$, $V$, $R$.

$$N = (0,0,1)$$

$$L = (4,5,3) - (1,2,3) = (3,3,0) \Rightarrow \left( \frac{3}{\sqrt{18}}, \frac{3}{\sqrt{18}}, 0 \right)$$

$$V = (1,2,6) - (1,2,3) = (0,0,3) \Rightarrow (0,0,1)$$

$$R = \text{reflected on } Z=3 \text{ plane } \Rightarrow \left( -\frac{3}{\sqrt{18}}, \frac{3}{\sqrt{18}}, 0 \right)$$

d) What is the color of the pixel at $x$, assuming the object has material properties $k_a=0.2$, $k_d=0.4$, $k_s=0.6$, $n_e=5$.

$$= 0.2 + 0.4(0.0,0.1) \cdot \left[ \frac{3}{\sqrt{18}}, \frac{3}{\sqrt{18}}, 0 \right] + 0.6 \left[ -\frac{3}{\sqrt{18}}, -\frac{3}{\sqrt{18}}, 0 \right] \cdot \left[ 0,0,1 \right]$$

$$= 0.2 + 0.4(0) + 0.6(0)$$

$$= 0.2$$

e) Suppose that we now wanted to render a retroreflective object, one for which most of the light returns in the direction of the light source rather than in the reflected direction. The existing OpenGL lighting equation doesn’t really deal with this case. Suggest a 4th portion of the equation that could be added to the equation you wrote down in part (a).

Instead of waiting for specular when the eye is near the reflection direction, we now want a highlight when the eye is near the light direction.

$$k_e (L \cdot V)^n$$
5. What is the matrix necessary to perform the following 2D transformation? You don’t need to actually solve the math, just set up far enough that I would get an actual matrix result if the math was completed.

\[ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = m \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \]

(a) Right down the matrix, M, in symbolic form using \( R(\text{degrees}) \), \( T(x,y) \), \( S(x,y) \), but with values for \( \text{degrees} \), \( x \), \( y \). (Be careful about left or right multiplication.)

\[ M = T(4,3) \circ S(1,2) \]

(b) Now right down the actual matrix elements to build the 3x3 matrices you specified above. (Its fine to leave math and sine, cosine, etc in your individual elements, but they should be sufficiently specified to evaluate to numbers if we have a calculator)

\[ M = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \]

(c) If I multiply the following points against the matrix M you specified above, what is the result?

\( (0,0) \rightarrow (4,3) \)
\( (2,2) \rightarrow (6,7) \)
6. Raytracing

Complete the primary eye rays, and draw the shadow and reflection rays for this scene, labeling them with shadow or reflected ray. For each primary eye ray, label the final pixel color. The mirror components reflect all light in RGB. The diffuse components have color as indicated.

Colors of rays = 3 points
Shadow rays in right places = 3 points
Reflection rays in right places = 4 points