Computer Graphics - CS562-402

Madjid Allili ~ Johnson 125 ~ ext. 2740 ~ malllili@ubishops.ca

Midterm Exam - Solutions

1. (a) Find the matrix that converts the coordinates $\mathbf{a}^T = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ of a point P in the frame $[u_1, u_2, u_3, P_0]$ into coordinates $\mathbf{b}^T = [\beta_1, \beta_2, \beta_3, \beta_4]$ in the new frame $[v_1, v_2, v_3, Q_0]$, given by

$$v_1 = u_1, \quad v_2 = u_2, \quad v_3 = u_2 + u_3, \quad Q_0 = P_0 + u_1$$

(b) Specify the vector $\mathbf{b}$ if $\mathbf{a}^T = [1,\ 1,\ 1,\ 1]$.

ANSWER:

(a) The matrix that converts the coordinates $\mathbf{a}^T = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ of a point P in the frame $[u_1, u_2, u_3, P_0]$ into coordinates $\mathbf{b}^T = [\beta_1, \beta_2, \beta_3, \beta_4]$ in the new frame $[v_1, v_2, v_3, Q_0]$ is $\left(M^T\right)^{-1}$ where

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

It follows that

$$\left(M^T\right)^{-1} = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) If $\mathbf{a}^T = [1,\ 1,\ 1,\ 1]$ then

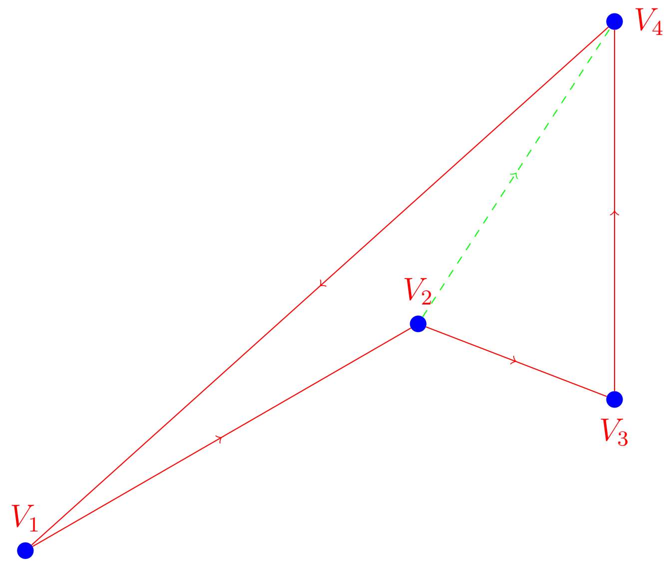
$$b = \left(M^T\right)^{-1} a = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

---

2. Explain how we can determine whether the following polygon specified by the ordered set of vertices in counterclockwise orientation

$$P = \{(2,1),\ (6,4),\ (8,3),\ (8,8)\}$$

is convex or concave. If the polygon is determined to be concave by this method, how can it then be split into a set of convex polygons?



Starting from vertex V_1, we compute cross products of consecutive edges in the counter clockwise orientation, we get a negative result only for edges V_1V_2 and V_2V_3 (See Homewrok#2, exercise 8). This means the polygon is concave and the problem occurs at vertex V_2. Generally speaking, we can start subdividing the polygon by adding and edge starting at V_1 and joining an existing vertex or a new vertex and then continue checking for convexity. In our situation, by adding the edge V_2V_4, we subdivide the polygon into two triangles which are necessarily convex.

3. A pair of transformations is said to commute if the order in which you apply them does not matter. In terms of transformation matrices, that means that $AB = BA$. Explain which of the following transformation pairs commute in 3D.

   (a) translate - translate

   (b) scale - scale

   (c) rotate - translate

(d) rotate - scale

(e) rotate - rotate

ANSWER:

(a) A translate and a translate commute:

$$\begin{pmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & x_2 \\ 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & z_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & x_1 + x_2 \\ 0 & 1 & 0 & y_1 + y_2 \\ 0 & 0 & 1 & z_1 + z_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) A scale and a scale commute:

$$\begin{pmatrix} r_1 & 0 & 0 & 0 \\ 0 & s_1 & 0 & 0 \\ 0 & 0 & t_1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_2 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & t_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_1 r_2 & 0 & 0 & 0 \\ 0 & s_1 s_2 & 0 & 0 \\ 0 & 0 & t_1 t_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(c) A rotate and a translate do not commute. For example, a rotate about z axis:

$$\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & x\cos\theta - y\sin\theta \\ \sin\theta & \cos\theta & 0 & x\sin\theta + y\cos\theta \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The two results are different.

(d) A rotate and scale do not commute. For example, a rotate about z axis:

$$\begin{pmatrix} r & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r\cos\theta & -r\sin\theta & 0 & 0 \\ s\sin\theta & s\cos\theta & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r\cos\theta & -s\sin\theta & 0 & 0 \\ r\sin\theta & s\cos\theta & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The two results are different.

(e) A rotate and rotate do not commute in general (unless, they are performed about the same axis - about the origin in 2D -):

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \cos\phi\sin\theta & \cos\phi\cos\theta & -\sin\phi & 0 \\ \sin\phi\sin\theta & \sin\phi\cos\theta & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta\cos\phi & \sin\theta\sin\phi & 0 \\ \sin\theta & \cos\theta\cos\phi & -\cos\theta\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The two results are different.

---

4. (a) Find the matrix for mirror reflection with respect to the plane passing through the origin and having a normal vector $\mathbf{n} = (1, 1, 0)$.

Two methods to get the matrix.

i. You may consider the matrix of the axonometric projection on the plane through the origin with normal vector $\mathbf{n} = (1, 1, 0)$. One can remark that $\mathbf{n} = (1, 1, 0) = \sqrt{2}\mathbf{N} = \sqrt{2}(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$, where $\mathbf{N}$ is a unit normal vector to the plane.

Let (x_p, y_p, z_p) be the projection of (x, y, z) on the plane. The coordinates (x', y', z') of the reflection of (x, y, z) with respect to the plane above satisfy the relations

$$x_p = \frac{x + x'}{2}, \qquad x_p = \frac{y + y'}{2}, \qquad z_p = \frac{z + z'}{2}.$$

This follows that

$$x' = 2x_p - x, \qquad y' = 2y_p - y, \qquad z' = 2z_p - z.$$

Since the axonometric projection matrix preserves the homogeneous fourth coordinate 1, we can see that

$$(x', y', z', 1) = 2(x_p, y_p, z_p, 1) - (x, y, z, 1)$$

that is

$$M = 2A_p - I.$$

where M is the mirror reflection w.r.t. the plane, A_p is the axonometric projection matrix on the plane, and I the 4×4 identity matrix. Thus

$$M = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ii. The second method is to align $\mathbf{n}$ with $\mathbf{k}$, reflect about the xy-plane, and then undo the first transformation. Thus

$$\begin{aligned} M &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

(b) Find the image of the pyramid defined by the vertices $A(0,0,0)$, $B(1,0,0)$, $C(0,1,0)$ and $D(0,0,1)$ by this transformation.

$$M(ABCD) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

---

5. Given the following code, find the current `Modelview` matrix.

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatei( 5, 2, 1 );
glRotatef( 45.0f, 0.0, 0.0, 1.0f );
```

The Modelview matrix M is given by the 45^0-rotation around the z-axis followed by the translation by vector $(5, 2, 1)$. Hence,

$$M = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\pi/4 & -\sin\pi/4 & 0 & 0 \\ \sin\pi/4 & \cos\pi/4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\pi/4 & -\sin\pi/4 & 0 & 5 \\ \sin\pi/4 & \cos\pi/4 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

---

6. Figure 1 displays the result of the function `draw_house()`.

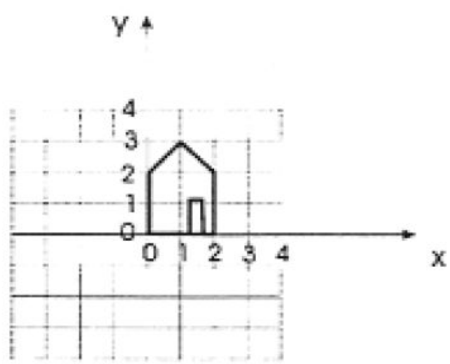


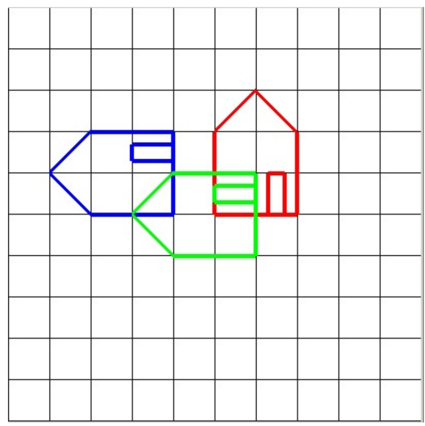
Figure 1: `Home sweet home`.

(a) Draw the pictures of house A and house B which are the transforms of the original house by the following OpenGL commands.

```
glTranslate(1,0,0);
glRotate(90,0,0,1);
glPushMatrix();
glTranslate(0,2,0);
```

```
draw_house();
glPopMatrix();
glTranslate(-1,0,0);
draw_house();
```

Answer:



(b) Give the series of affine transformations (assuming post-multiplying) needed to create the picture in Figure 2, assuming the house started from the position shown in Figure 1.

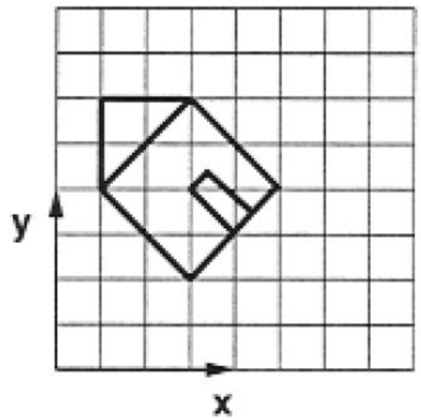


Figure 2: Transformed sweet home.

ANSWER:

```
glTranslate(3,2,0);
glRotate(45,0,0,1);
glScale(√2,√2,1);
draw_house();
```

7. Give a sequence of the following commands for drawing the scene depicted in the Figure 3 below. Assume that the current transformation matrix is initialized to the identity. All transformation matrices are multiplied to current transformation matrix from the right:

   `drawSquare()`: draw a 1×1 square, bottom left corner at origin
   `translate( x, y )`: translation by x, y
   `scale( x, y )`: scaling in x and y direction
   `rotateZ(`θ` )`: rotation around z axis (i.e. in $x - y$ plane)
   `pushMatrix()`: duplicates top entry of matrix stack
   `popMatrix()`: deletes top entry of matrix stack.

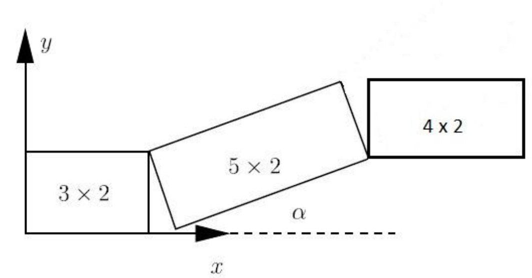


Figure 3: Hierarchical Transformations.

Answer:

```
pushMatrix();
scale(3,2);
drawSquare();
popMatrix();
translate(3,2);
rotateZ(α);
translate(0,-2);
pushMatrix();
scale(5,2);
drawSquare();
popMatrix();
translate(5,0);
rotateZ(-α);
pushMatrix();
scale(4,2);
drawSquare();
popMatrix();
```