SOS Graphics Problems Week 6 - Name(s) **SOLUTIONS**

Hand these in Lab Week 6, or in Class Week 7; just do them paper and pencil….

1) **3D Transforms:** What is the 4x4 matrix transform (in homogeneous coordinates) for the 3D transformations below. *Also give the inverse*.

a) Scale by 5 in the z direction:

The transform:	The inverse:
1 0 0 0 0 1 0 0 0 0 5 0 0 0 0 1	1 0 0 0 0 1 0 0 0 0 1/5 0 0 0 0 1

b) A rotation of 10 degrees about the x axis:

The transform:	The inverse: replace 10 with -10 to give:
1 0 0 0 0 cos(10) −sin(10) 0 0 sin(10) cos(10) 0 0 0 0 1	1 0 0 0 0 cos(10) sin(10) 0 0 −sin(10) cos(10) 0 0 0 0 1

c) A projection onto the yz-plane.

The transform:	The inverse:
0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	no inverse exists.

d) A translation by 10 along x and by -5 along y.

The transform:	The inverse:
1 0 0 10 0 1 0 -5 0 0 1 0 0 0 0 1	1 0 0 −10 0 1 0 5 0 0 1 0 0 0 0 1

e) A reflection through the xz-plane

The transform:	The inverse:
1 0 0 0 0 −1 0 0 0 0 1 0 0 0 0 1	1 0 0 0 0 −1 0 0 0 0 1 0 0 0 0 1

2) **Composition of 3D Transforms:** What is the sequence of transformations needed to achieve the operations given below. Also, include the corresponding inverse. (fyi, we give the 4x4 matrices below – so you can see the correspondence), but you did not need to write out the 4x4 matrices; instead, you were to make use of the syntax (which is closer to what you do when writing code):

Scale: $S(s_X,s_Y,s_Z)$
Translation: (t_X,t_Y,t_Z)
Rotation: $R_X(\Theta)$, $R_Y(\Theta)$, $R_Z(\Theta)$.

a) A rotation of 20 degrees about an axis that goes through the point (a,b,c) and is parallel to the y axis.

The transforms:

	Rotate(r0, r(1/9pi), r0) * Translate(ta,tb,tc)	
	Rotate	Translate
$T(a,b,c)\ R_y\ (20)\ T(-a,-b,-c)$	cos(1/9pin) 0 sin(1/9pin) 0 0 1 0 0 -sin(1/9pin) 0 cos(1/9pin) 0 0 0 0 1	1 0 0 a 0 1 0 b 0 0 1 c 0 0 0 1

The inverse:

	Rotate(r0, r(-1/9pi), r0) * Translate(t-a,t-b,t-c)	
	Rotate	Translate
$T(a,b,c)\ R_y\ (-20)\ T(-a,-b,-c)$	cos(1/9pin) 0 -sin(1/9pin) 0 0 1 0 0 sin(1/9pin) 0 cos(1/9pin) 0 0 0 0 1	1 0 0 -a 0 1 0 -b 0 0 1 -c 0 0 0 1

b) A scale by 5 (with fixed point at the origin) along the direction defined by the line from (0,0,0) to (-1, 0, 1).

The transforms:

<table>
<tr><td>$R_y(45)$ S(5,1,1) $R_y(-45)$
or
$R_y(-45)$ S(1,1,5) $R_y(45)$</td><td>Scale(s(5/rt2), s1, s(5/rt2))
<table><tr><td><u>Scale</u>
5/rt2 0 0 0
0 1 0 0
0 0 5/rt2 0
0 0 0 1</td><td><u>Scale</u>
rt2/5 0 0 0
0 1 0 0
0 0 rt2/5 0
0 0 0 1</td></tr></table></td></tr>
</table>

The inverse:

<table>
<tr><td>$R_y(45)$ S(1/5,1,1) $R_y(-45)$
or
$R_y(-45)$ S(1,1,1/5) $R_y(45)$</td><td>Scale(s(5/rt2), s1, s(5/rt2))
<table><tr><td><u>Scale</u>
5/rt2 0 0 0
0 1 0 0
0 0 5/rt2 0
0 0 0 1</td><td><u>Scale</u>
rt2/5 0 0 0
0 1 0 0
0 0 rt2/5 0
0 0 0 1</td></tr></table></td></tr>
</table>

c) A scale by 2 with fixed point (2,3,4) and along the direction parallel to the x axis.

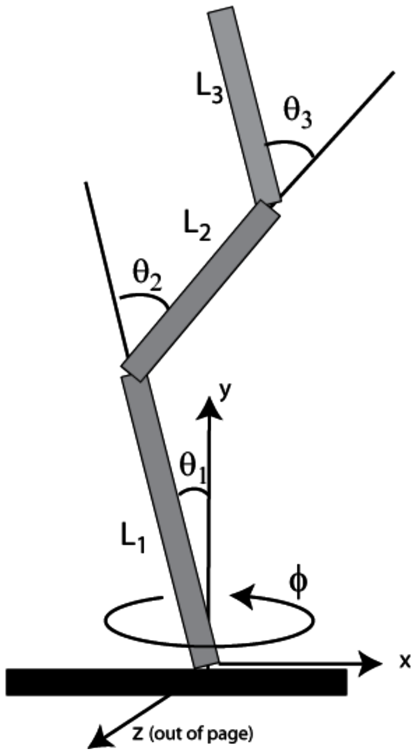
The transforms:

<table>
<tr><td>T(2,3,4) S(2,1,1) T(-2,-3,-4)</td><td>S(s5,s1,s1) * T(t2,t3,t4)
<table><tr><td><u>Scale</u>
2 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1</td><td>Translate
1 0 0 2
0 1 0 3
0 0 1 4
0 0 0 1</td></tr></table></td></tr>
</table>

The inverse:

<table>
<tr><td>T(2,3,4) S(1/2,1,1) T(-2,-3,-4)</td><td>S(s.5,s1,s1) * T(t-2,t-3,t-4)
<table><tr><td><u>Scale</u>
.5 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1</td><td><u>Scale</u>
1 0 0 -2
0 1 0 -3
0 0 1 -4
0 0 0 1</td></tr></table></td></tr>
</table>

3) **Scene Graphs:** Below is a picture of a 3 segment robotic arm sitting on a base. Each segment is a cylinder of radius r and length L_i, with i=1,2,or 3. The arm segments can be rotated as shown.



Draw the <u>scene graph</u> for the robotic arm (not including the black base).

Assume that you have access to a cylinder primitive that has radius 1, height 1, is centered at the origin, and aligned with the z-axis.

Be sure to include all transformations. Scale transformations should be indicated as **S(s_x,s_y,s_z)**where you fill in specific values for s_x, s_y, and s_z. Similarly, translations and rotations should have the form **T(t_x,t_y,t_z), Rx(angle), Ry(angle),** and **Rz(angle).**
Indicate push/pops where needed.

NOTE: the code for this can be found on the fileshare at: https://myfiles.evergreen.edu/academics/programs/sosoftware/Handouts/Graphics/Lectures/Lecture08/