October 14, 2014 Name ___________________

CS445 Midterm *Solutions*

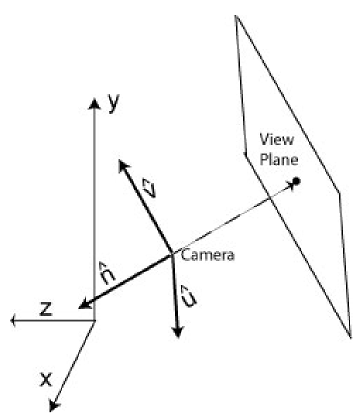
Fall 2014

1.	(max = 12)	5.	(max = 18)
2.	(max = 4)	6.	(max = 16)
3.	(max = 10)	7.	(max = 20)
4.	(max = 20)		
Final Score:		(max=100)	

Please try to write legibly. The instructor's failure to decipher what you write will be considered an incorrect answer.

1) (12 pts) Ray Tracing: Suppose you are given:
   - VPN = a vector that points in a direction *opposite* the way the camera looks
   - VUP = the up direction vector

   How does one calculate the normalized eye coordinate basis vectors: $\hat{u}, \hat{v}, \hat{n}$ (see picture). Assume you are using a <u>*right handed*</u> coordinate system as shown.



$$\hat{u} = \frac{VUP \times \hat{n}}{\|VUP \times \hat{n}\|}$$

$$\hat{v} = \hat{n} \times \hat{u}$$

$$\hat{n} = \frac{VPN}{\|VPN\|}$$

2) (4 pts) A plane can be described by a point on its surface (Q) and a vector (N) which is normal to its surface. How do you determine mathematically whether an arbitrary point (P) is or is not on the plane?

   If $(P - Q) \cdot N = 0$
   then P is in the plane.
   Otherwise, it isn't.

3) (2 pts each, 10 pts total) Multiple Choice – Circle the correct answer:

a) The cross product (0, 1, 0) X (1, 0,0) is

i) (0,0,1)
ii) (1,1,0)
iii) (0,0,-1)
iv) (1,1,-1)
v) None of the above.

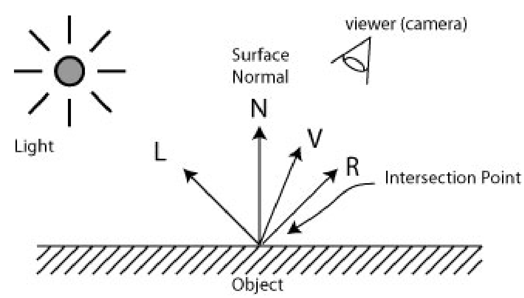
b) The dot product of two vectors is largest when the vectors are oriented in directions which are:

i) Perpendicular
ii) 180 degrees apart (i.e. opposite)
iii) 45 degrees apart
iv) 0 degrees apart (i.e. parallel)
v) None of the above.

c) In homogeneous coordinates, the expression (x,y,z,w) with w = 1 represents

i) Nothing, it is undefined
ii) Point
iii) Ray
iv) Vector
v) None of the above.

d) In the phong lighting model, the diffuse light component depends on the vector product



i) N • V
ii) N x L
iii) L • N
iv) V • R
v) None of the above.

e) In the phong lighting model, reflections are obtained by tracing a ray in a direction calculated from

i) L and N
ii) R
iii) V and N
iv) L and V
v) None of the above

4) (5 pts each, 20 pts total) **3D Transforms:** What is the 4x4 matrix transform (in homogeneous coordinates) for the 3D transformations below. *Also give the inverse*.

a) A uniform scale by 2 about the origin:

The transform:

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse:

$$\begin{pmatrix} .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b) A rotation of 60 degrees about the y axis:

The transform:

$$\begin{pmatrix} \cos(60) & 0 & \sin(60) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(60) & 0 & \cos(60) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse:

$$\begin{pmatrix} \cos(60) & 0 & -\sin(60) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(60) & 0 & \cos(60) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

c) A reflection through the xz-plane.

The transform:

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

The inverse:

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

d) A translation by 2 along x and by 4 along z.

The transform:

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse:

$$\begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5) (6 each, 18 pts total) **Composition of 3D Transforms:** What is the sequence of transformations needed to achieve the operations given below. Also, include the corresponding inverse. You do not need to write out the 4x4 matrices. Instead, make use of the syntax:

Scale: $S(s_x,s_y,s_z)$
Translation: $T(t_x,t_y,t_z)$
Rotation: $R_x(\Theta)$, $R_y(\Theta)$, $R_z(\Theta)$.

a) A scale by 10 along the z-axis about the point (a,b,c).

The transforms:

T(a,b,c) S(1, 1, 10) T(-a,-b,-c)

The inverse:

T(a,b,c) S(1, 1, 1/10) T(-a,-b,-c)

b) A rotation by 10 degrees about an axis defined by the line from (0,0,0) to (0, 1, 1).

The transforms:

Rx(45) Ry(10) Rx(-45)

or

Rx(-45) Rz(10) Rx(45)

The inverse:

Rx(45) Ry(-10) Rx(-45)

or

Rx(-45) Rz(-10) Rx(45)

c) A scale by 5 with fixed point (2,3,4) and along a direction defined by the line from (0,0,0) to (1, 0, 1).

The transforms:

T(2,3,4) Ry(-45) S(5,1,1) Ry(45) T(-2,-3,-4)

or T(2,3,4) Ry(45) S(1,1,5) Ry(-45) T(-2,-3,-4)

The inverse:

T(2,3,4) Ry(-45) S(1/5,1,1) Ry(45) T(-2,-3,-4)

or T(2,3,4) Ry(45) S(1,1,1/5) Ry(-45) T(-2,-3,-4)

6) (16 pts total) Given the following sequence of commands in openGL:

```
Line  0: mvMatrixStack.loadIdentity();
Line  1: mv = mv*Translate(x,y,z);
Line  2: mvMatrixStack.pushMatrix(mv);
Line  3: mv = mv*RotateX(90);
Line  4: glUniformMatrix4fv( model_view, 1, GL_TRUE, mv );
Line  5: shapes.drawCylinder();

Line  6: mvMatrixStack.pushMatrix(mv);
Line  7: mv = mv*Scale(2,3,2);
Line  8: glUniformMatrix4fv( model_view, 1, GL_TRUE, mv );
Line  9: shapes.drawDisk();

Line 10: mv = mvMatrixStack.popMatrix();
Line 11: mv = mv*RotateZ(180);
Line 12: glUniformMatrix4fv( model_view, 1, GL_TRUE, mv );
Line 13: shapes.drawDisk();

Line 14: mv = mvMatrixStack.popMatrix();
Line 15: mv = mv*Scale(2,4,4);
Line 16: glUniformMatrix4fv( model_view, 1, GL_TRUE, mv );
Line 17: shapes.drawCube();
```

a) (8 pts) What is the model view matrix `mv` (i.e. *expressed as a product of transformations*) at lines 5, 9, 13, and 17 (i.e. at the point when shapes are drawn). Assume `mv` is initialized to be the identity.
For brevity, use the same syntax as in problem 5.:

Line 5: mv = `T(x,y,z) Rx(90)`

Line 9: mv = `T(x,y,z) Rx(90)S(2,3,2)`

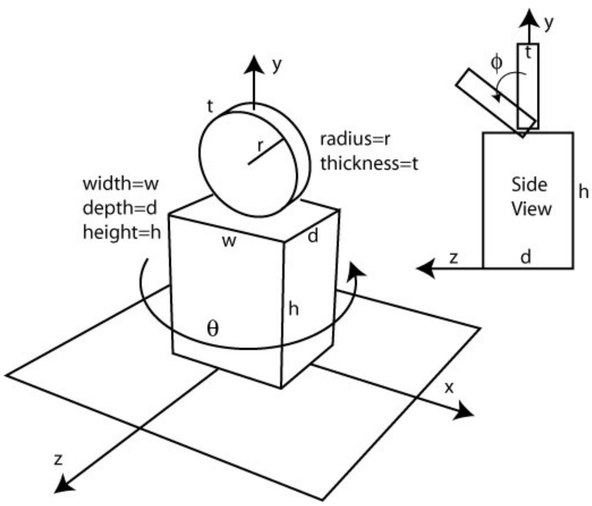
Line 13: mv = `T(x,y,z) Rx(90) Rz(180)`

Line 17: mv = `T(x,y,z)S(2,4,4)`

b) (8 pts) What is stored on the matrix *stack* at these same lines?

Line 5	Line 9	Line 13	Line 17
`T(x,y,z)`	`T(x,y,z)Rx(90)` `T(x,y,z)`	`T(x,y,z)`	`(nothing)`

7) (20 pts) **Scene Graphs:** Below is a picture of a robot sitting on a ground plane. The body is a rectangular box with dimensions (w,h,d), as indicated. The head is a cylindrical shape with a thickness t and radius r. The entire robot can rotate around the y axis by angle ϴ and translate as though it is walking *forward* (in the picture, this would be in the z direction, but if first turned by ϴ=90 degrees, it would be moving along x). The head moves with the body but can also rotate, relative to the body, by ϕ about the neck point (see side view).



Draw the scene graph for this robot (not including the ground plane).

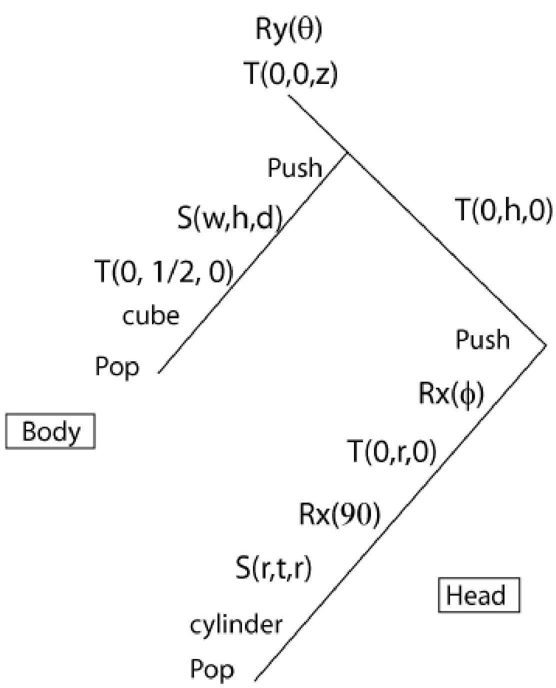
Assume that you have access to

- *a capped cylinder primitive with radius 1, height 1, centered at the origin, aligned with the y-axis.*
- *A cube primitive with side 1, centered at the origin.*

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.

It is recommended that you first work it out on scratch paper.

There are a number of ways one could have drawn the scenegraph. Here are two possibilities:



or