Written Assignment 1 Name____________

15-462, Spring 2007

Due in class Thursday 2/8 or in Jessica Hodgins's mailbox in NSH 4th floor by 9am on Friday 2/9. Please do each problem on a separate piece of paper.

1. **Graphics Pipeline**
   (a) Briefly describe the stages of the graphics pipeline.
   (b) The Z-buffer algorithm makes hidden surface removal simple but it makes it tricky to handle transparency. Why? How might you handle transparency?
   (c) Computers have gotten much faster over the past years. Therefore the rotating cube example we discussed in the class might not work the way authors intended it to work and the cube rotates quite fast. How can you make it rotate slower without adding unnecessary burden on the CPU?
   (d) Why does OpenGL maintain a large number of state variables (such as color, transform, etc)?
   (e) Why are display lists useful? How does it affect the transfer of data from the CPU to GPU?

Name____________

2. Vector Operations

Using only vector operations---subtraction, addition, scalar multiplication, dot product and cross product---compute answers to the following questions. You can also subtract points to compute a vector. For example (B-A) is a vector parallel to the line through points B and A. Assume right-handed coordinate system.

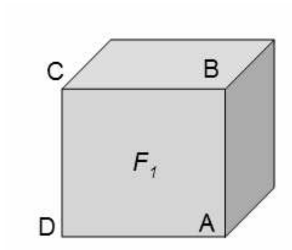

Figure 1: Cube with front face F_1 defined by 4 vertices: A,B,C,D.

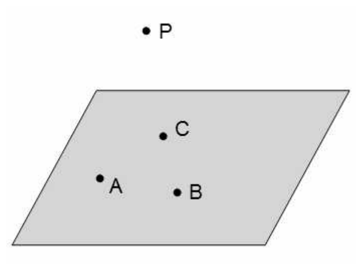

Figure 2: Plane in 3D defined by 3 points A, B and C.

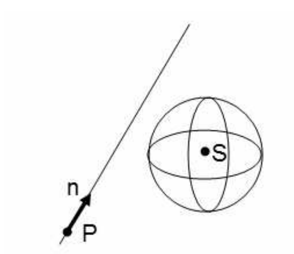

Figure 3: Line ***L*** in 3D defined by point ***P*** and unit vector ***n*** and sphere defined by center ***S*** and radius ***R***.

(a) Compute normal facing outside of the cube in ***Figure 1*** for face F_1.
(b) What convention does OpenGL use to determine the direction of a normal for a triangle when none is explicitly specified?
(c) Compute the distance from point ***P*** to the plane defined by 3 points ***A, B*** and ***C*** (***Figure 2***).
(d) Show how to determine if the line ***L*** defined by point ***P*** and unit vector ***n*** intersects a sphere with radius ***R*** and center ***S*** (***Figure 3***).

Name____________

3. Implicit and parametric equations

Question 11, Chapter 2 in Shirley Book.

(a) What is the implicit equation of the plane through 3D points (1,0,0), (0,1,0) and (0,0,1).
(b) What is the parametric equation?
(c) What is the normal vector to this plane?

Name____________

4. **2D transformations**

Assume the following notation: **R(ϕ)** --- homogeneous matrix that rotates counterclockwise around axis coming out of the (X,Y) plane by angle ϕ; **T (x,y)** --- homogeneous matrix that translates by ***x*** along the ***X*** axis and by ***y*** along the ***Y*** axis.

We want to draw a robot arm consisting of a hierarchy of three identical rectangles-A,B and C. Rectangle B is connected to rectangle A with a one degree of freedom joint and rectangle C is connected to rectangle B with a one degree of freedom joint. Initially, all three rectangles are located at the origin as shown in **Figure 4(a)**.

We want to compute the transformation matrix for each rectangle that would transform it into the desired configuration. For example, to transform rectangles into the configuration shown in **Figure 4(b)** we would compute the following matrices:

$\mathbf{M_A}$**= T (0, 0) ---** matrix to transform rectangle A.
$\mathbf{M_B}$**= T (0, 10) ---** matrix to transform rectangle B.
$\mathbf{M_C}$**= T (0, 20) ---** matrix to transform rectangle C.

(a) Using this notation, compute matrices $\mathbf{M_A}$, $\mathbf{M_B}$, $\mathbf{M_C}$ that would transform the rectangles into the configuration shown in **Figure 4(c).** Write each matrix as a composition of one translation and one rotation matrices.

(b) In (a) we transformed each rectangle independently. Now we will use a hierarchical approach. The goal is to compute the transformation matrix for the child segment using the transformation matrix for the parent segment and the local transformation of the child segment with respect to the parent segment. For the robot arm, the hierarchy is as follows: A is a parent of B and B is a parent of C. Therefore you can compute $\mathbf{M_A}$ as before. You will then need to derive the transformation matrix for segment B (**$\mathbf{M_B}$)** using the transformation matrix of its parent (**$\mathbf{M_A}$)** and the local transformation of segment B with respect to segment A. Similarly, derive $\mathbf{M_C}$ from $\mathbf{M_B}$. **Hint**: This derivation becomes much easier if instead of thinking in terms of global coordinate system, you imagine that a local coordinates system is tied to each rectangle and that all operations occur relative to this changing coordinate system (see OpenGL book, Chapter 3, "Thinking About Transformations" section).

(c) What are the advantages/disadvantages of (b) over (a). When might you want to use (b) instead of (a) and vice versa?

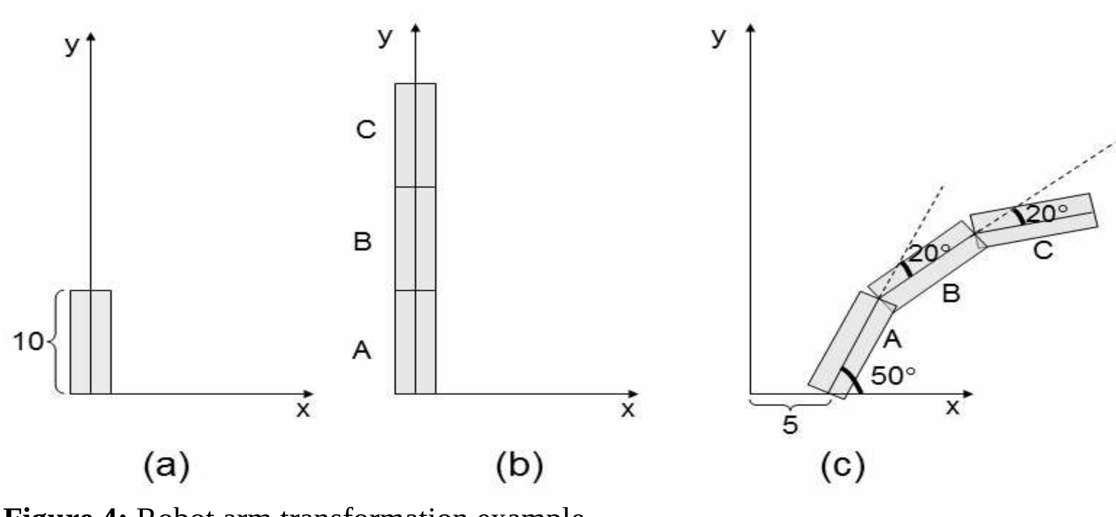

Figure 4: Robot arm transformation example.