Computer Graphics	**Homework 1v2**
CS417/CS418 Fall 1998	**Due Wednesday, February 10, 1999**

- Make sure you've turned in the waiver and its requested e-mail.
- Hand in each part (CS417, CS418, administrative) SEPARATELY.
- Make sure your names are legible and easily spotted on the front page of each part.
- As always in this class, you are graded on correctness *and also clarity and conciseness.*
- Write legibly & sign & date the following (fill blanks appropriately):
"I, ____, CUID# ____, wrote up this assignment; my partner is ____, CUID# ____."
- E-mail how many total hours this homework takes you by midnight, Friday, February 12.

0. Follow the instructions above. *Failure to do so will be penalized.* Feedback is always welcome.

CS417 Part

1. Let us look at two derivations of the distance between a point $w = (w_1, w_2)$ and the line defined by the origin and a point $v = (v_1, v_2) \neq (0, 0)$. Show all your work.

   (a) Compute the projection u of w onto v and then compute the distance between u and w. Stick with vector notation (no vector components).

   (b) Compute a (non-zero) vector u that is perpendicular to v (solve for $u \cdot v = 0$ or rotate v by ± 90 degrees). Compute the length of the projection of w onto u. For this part, expand your answer in terms of vector components.

2. Suppose we rotate, scale (uniformly in all directions), and translate the plane so that the the origin ends up at $a = (x_a, y_a)$, and the point $(1, 0)$ ends up at $b = (x_b, y_b)$. NOTE: No flipping/reflection (e.g. $(x, y) \mapsto (x, -y)$) is done.

   Express this sequence of transformations as a single 3-by-3 matrix (that is applied to homogenous coordinates). Show your work. Your final answer should not involve any angles or trig functions.

3. (Moved into CS418 part)

4. Write a Matlab function so that `[x,y,z]=torus(m,n,r,R); surf(x,y,z)` plots a torus obtained as follows. Place a circle in the xz-plane with center $x = R, z = 0$ and radius r; sample the circle at m points. Revolve the circle around the z-axis; sample at n points, i.e. create n such circles. Don't worry about accuracy (the original `(my)sphere` code worries that `cos(pi/2)` does not yield 0).

   Avoid loops. Demonstrate with $m = 11, n = 40, r = 1, R = 2$; use `axis equal` to fix the aspect ratio. Turn in your code and a printout of the plot.

   HINT: modify `mypshere`, which does the following. Place half a unit circle in the xz-plane centered at the origin, sampled at m points. Revolve the half circle around the z-axis, sampled at n points.

CS418 Part

5. (Moved out of HW#1)

3. Write a Matlab function so that `dst=rotatesuper(src,angle,n)` returns the source image `src` rotated by `angle` radians. For better accuracy than our code from lecture/newsgroup, divide each source pixel into $n \times n$ subpixels, map them to destination pixels, and average appropriately. Avoid loops. Turn in your code **and a printout of the image from `durer` rotated by $\pi/6$ radians with 2×2 super-sampling**. (If memory is a problem, then crop the original image to `X(200:400,200:400)`.)