

Problem Set 1
CMSC 427
Distributed Tuesday, February 8, 2005
Due: Tuesday, February 22, 2005

Programming

1. **20 points.** Modify the program from problem set 1 so that you can rotate the square. Remove the previous functionality of the right mouse, so that clicking it no longer creates a diamond. Instead, clicking the right mouse and holding it down should enable rotation about the center of the square. More precisely, let p_1 be the center point of the square. Let p_2 be the location of the mouse when you click on the right button. Let p_3 be the position the mouse is moved to while holding the right button. The square should be rotated by the angle formed by (p_2, p_1, p_3) . The rotation should be either clockwise or counterclockwise, depending on the direction the mouse was moved. When you release the mouse's right button, the rectangle should remain in its final position.

Hint: There are a number of reasonable ways to do this. One is to set up a new coordinate system when the right mouse is clicked, and then as it moves, determine its position relative to this coordinate system.

3D Motion: We are giving you skeleton code that will display a rectanguloid on the screen. You will enhance this code to add more rectanguloids, to change the type of projection, and to allow yourself to navigate about the scene. Note that included in the code that we give you is a file `vector3.cpp` that contains many useful functions for vectors, including functions to perform rotations.

2. **10 points.** Implement keyboard interaction that allows you to change the projection mode. Typing 'p' should change projection to perspective. Typing 'o' should change back to orthographic.
3. **10 points.** Add four more rectanguloids to the scene, each with a different color, shape, and position.
4. **10 points** Implement a mouse control to allow you to move forward or back in the scene. Holding down the left mouse button should move you forward slowly. Holding down the right mouse should move you back. Holding both buttons should cause no movement. Throughout the assignment, do not worry about what happens if your motion takes you through a rectanguloid.
5. **10 points.** Now implement rotations about your Up axis. If any mouse buttons are being held, moving the mouse to the left or right should cause your viewpoint to rotate about the viewer's Up axis by an amount proportional to the distance the mouse was moved. This action should interact properly with part 4 - that is, after you have rotated, the motion will be in the direction that you are currently pointing.
6. **10 points.** Now implement a second rotation. Moving the mouse forward will cause you to rotate up. More precisely, your rotation will be about an axis that is orthogonal to the Up axis and to the viewing direction. Moving the mouse back will cause you to rotate down. Again, this should interact properly with the rest

of your controls. Horizontal mouse motions should continue to cause you to rotate about the Up axis. Holding a mouse button down should cause you to move forward or backward in the direction that you are currently facing.

Pencil and Paper. In the following problems we will ask you to provide some transformation matrices. It is fine to leave your final answer in the form of the product of several matrices; you don't need to multiply these out.

1. 2D Rotation: 10 points.

- Write down a matrix that will have the effect of rotating points about the origin counterclockwise by 30 degrees.
- Provide a matrix that we can apply to points that are written in homogenous coordinates so that they rotate by 30 degrees and then translate by (3,7).
- Provide a matrix that will rotate all points by 30 degrees about the point (3,7). That is, for example, this matrix should leave the point (3,7) right where it is.

2. 3D Rotation: 10 points.

- Provide a matrix that will rotate points about the y axis by 30 degrees (do not worry about whether it is clockwise or counterclockwise, either way is acceptable).
- Provide a matrix that will rotate the point (1,1,1) so that it now lies on the z axis.
- Provide a matrix that will rotate points by 30 degrees about the direction (1,1,1). For example, for any value of a this rotation should leave any point (a,a,a) unchanged.
- Provide a matrix that will rotate by 30 degrees points about an axis in the direction (1,1,1) that goes through the point (6,2,4). You should be able to apply this matrix to points written in homogenous coordinates. For example, this matrix should leave the point (6,2,4) unchanged, and also points (7,3,5), (8,4,6),

3. Changes of Viewpoint: 10 points. Suppose we are looking at the world along the direction (1,2,3). We want to describe the world with coordinates that fit the orientation of our camera. So our new z axis should be in the direction (1,2,3).

- Consider the point that had coordinates (17,9,4) in the old coordinate system. What would be the z coordinate of this point in the new coordinate system?
- Find possible directions for the x and y axis that are perpendicular to the new z axis and also to each other.
- Find a matrix that will convert the coordinates of a point in the old coordinate system to coordinates in this new coordinate system.

4. Challenge Problem: Determine the axis of rotation for the following rotation matrix:

$$\begin{pmatrix} 0.4505 & .2172 & .8660 \\ -0.2169 & .9675 & -0.1298 \\ -0.8660 & -.1294 & 0.4830 \end{pmatrix}$$