CS 535 Fall 2006

Assignment 1—due Thursday August 31st, 7AM

In a nutshell, this assignment asks you to implement rotation about an arbitrary axis and to lay the foundations of your future 3D graphics rendering system. Please start early (right away). You have a lot to do

Part A—rotation about an arbitrary axis

Given point *P* with coordinates (10, 3, 4) and axis *a* defined by point *A*(5, 0, 3) and direction vector *d*(-1/sqrtf(3), 1/sqrtf(3), -1/sqrtf(3)), rotate *P* with 1 degree increments about *a* until it reaches its original position. Save the coordinates of the rotated point in a text file. Using the 2D graph tool of your choice (e.g. Matlab, Excel, Xgraph) show the coordinates of the point as it is being rotated (a single graph with three curves, the rotation angle as the abscissa).

I recommend that you implement your own 3D vector and 3x3 matrix classes, however you are free to use any library you can find. Of course, you cannot use the rotate about arbitrary axis feature of a library. I will provide sample code for matrix inversion (email me for it).

Part B—towards a 3D graphics rendering system

Implement an application that

- Loads an image file (use a popular format such as tif (preferred) or jpeg, png, bmp etc)
- Displays the image file
- Allows the user to modify the image both using the mouse and the keyboard (be creative, examples of modifications include drawing 2D segments, circles, writing letters one pixel at a time, etc.)
- Saves the image (in the original format)

You can use any operating system, programming language, graphics API, graphics utility kit, and image formats. I will be using MS Visual C++, OpenGL, FLTK, and tiff in class. Here are links to the libraries. You are responsible for installing them on your computers and to learn how to use them. Ample documentation is provided. Ask questions. Carbon-copy the class alias.

http://www.fltk.org/
http://www.opengl.org/
http://www.libtiff.org/

Extra credit

- 3D visualization of the rotating point (2%)
- Show that expressing a direction vector in a new coordinate system does not use the origin of the new coordinate system (1%)
- Make a tile puzzle for Part B: subdivide the image in tiles, take one tile out, the computer shuffles the tiles, and the user is asked to recreate the original image by reordering the tiles; a tile can only move to the empty slot (2%)
- Make a "regular" puzzle for Part B: subdivide the image into puzzle pieces, shuffle them, and ask the user to put them back together; get the subdivision pattern from a real world puzzle (e.g. using a scanner), or create your own (4%)

What to turn in

- Turn in your assignments via the web; email me and the TA a URL with your assignment archive; let us know if you do not have access to web space
- Implementation source files and project; use relative paths; we should be able to build your project easily
- A report that includes the graph for part A and the output image for part B, as well as instructions on how to use the Graphical User Interface (GUI) for part B

Voicu Popescu, popescu@cs.purdue.edu , Fall 2006.