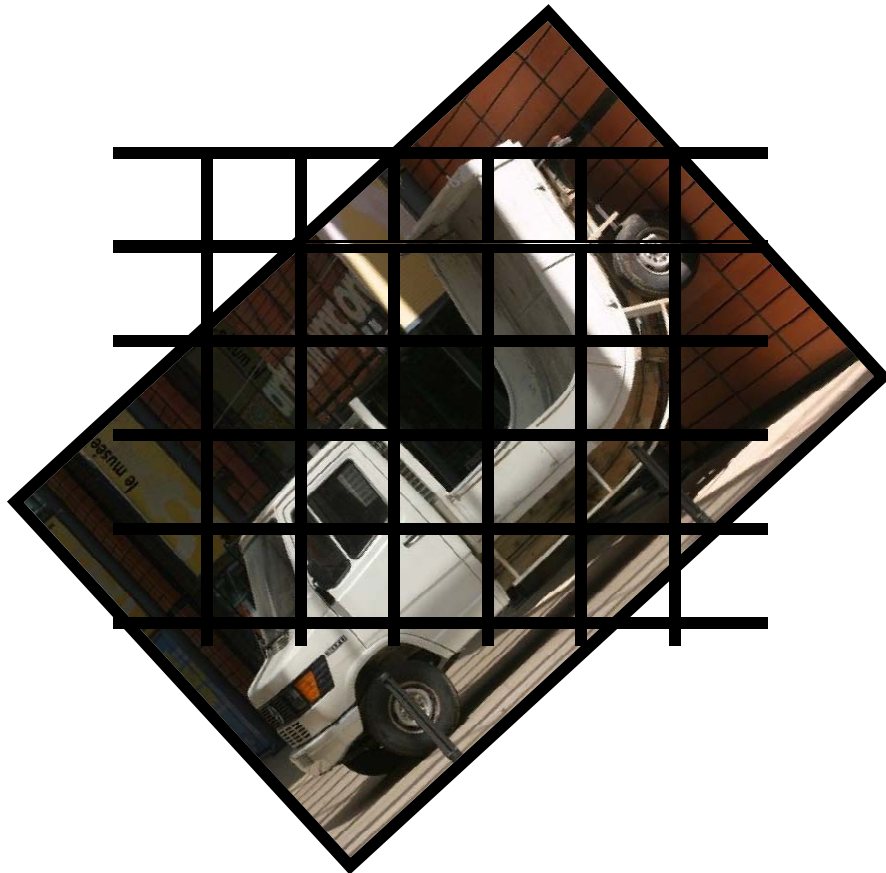


HOMEWORK #1

ABSTRACT. In this exercise you will write a program that animates a picture rotating on the screen. The exercise is slightly more involved for graduate students, for which the image should also scale up and down. The purpose of this assignment is to gain some familiarity in working with OpenGL and to apply operations to buffers of pixels representing images. It also introduces 2D linear transformations.

Due Feb 2, 2012, at 11:59pm.

- (1) Load an image from the file `pic1.jpg` and draw it into the window. The center of the image should be aligned with the center of the display window at all times. Next, animate the image so that it rotates perpendicular to the view axis, either clockwise or counter-clockwise, as shown in the figure below. The image should finish a full cycle every few seconds. For simplicity, you can fix the picture aspect ratio at 1:1 with a resolution of 600×600 pixels.



- (2) **For undergrads:** The display window should be larger than the image (in its dimensions) so the image can make a full cycle and be seen at all the times (i.e. no pixels are cropped out).
- (3) **For grads:** While rotating, the image should also shrink (scale down) and expand. It should reach each maximum size every fourth cycle, and its minimum size every fourth cycle. In its maximum size, the image should be 4 times larger than the the window. This means that if the width of both the image and the window are both 600×600 pixels, then when the image is in its max size only roughly 150×150 image pixels are visible in the window. When the image is in its minimal size, then most of the window pixels are black, and the image is presented in a small region of about 150×150 pixels.
- (4) In this assignment, you should use only the 2D rendering capabilities of OpenGL. In particular, you can initialize the drawing canvas using the function *gluOrtho2D* and then paint pixels into the window canvas using *glVertex2i*. You should handle the math of the 2D transforms yourself, in your own code. Most other GL functions, such as its built-in texturing capabilities, will be off-limits for this assignment. When in doubt, ask your TA!
- (5) The course webpage will have a link to a demo program ("paintdemo") which creates a simple GLUT window and paints pixels into it. It also includes example C code for decompressing a JPEG file into a buffer of pixels. Feel free to use this code as a starting point for your assignment.
- (6) Pay attention to how the colors of image pixels are mapped to the pixels in the window. This is where the 2D transformation will come into play. Do not leave pixels in the display window whose color is not assigned.
- (7) Your program should be reasonably efficient. So avoid unnecessary nested loops.
- (8) Your project should contain the following items:
 - (a) Your code files and a Makefile
 - (b) pic1.jpg, a picture chosen by you, which your program will read from the same directory.
 - (c) A readme, containing your name and email, known errata, and a summary of any additional features you may want considered for extra credit.
- (9) Submit the above items in a gz using the **D2L** system. The turnin name for this project is HW1.
- (10) Your submission must run on one of the graphics machines in GS930. The grader should be able to compile and link your project simply by typing "make" in the project directory.
- (11) Please check the course newsgroup and webpage regularly for updates or clarifications to this assignment.