

Assignment 0

Graphics Warmup

Overview:

This assignment is intended to be a gentle introduction to graphics programming. The goal is to write a very simple application which uses the basic features of the OpenGL, and some simple interaction with what is drawn on the screen.

The application will draw a single square in the center of a window, and should allow a user to resize, move and rotate this square. The look of the square should optionally change based on the point clicked.

Required Feature:

- Draw a square
- Allow the user to reposition the square
- Allow the user to resize the square
- Allow the user to rotate the square
- Provide a means to toggle the look of the square
 - I suggest changing the square's color as an easy option, but displaying a texture would be more ambitious
- Properly handle window resizing

Extra Credit:

Up to 20 points of extra credit are available for interesting extensions. Please write a paragraph explaining what you did. Here are some examples:

-Arbitrary polygons

-Visual indicator where to grab or what motion is being performed

-Animating squares

Submission Details:

Submissions should be in a form of a link to a webpage. The webpage should contain a .zip file containing your buildable code and an executable. Images of your working system are optional, but appreciated. I have limited inbox space: Do not send me an attachment!

Make sure your code compiles, I will **not** modify it to get it working once I receive it!

All assignments must contain a brief README.txt that contains:

- 1) Brief instructions for running your code
- 2) The honor code statement (typed out)
- 3) How many of your 5 allotted late days you are using

Simply submitting a picture of a solid color square in an “interesting” position will not count as a valid entry for the art contest. You will likely have to implement some extra credit feature to enter. Place your art contest submission as in image on your submission webpage.

Getting Started:

A .zip file with skeleton C++ code is online at:

<http://wwwx.cs.unc.edu/~sjguy/comp575/SquareSkeleton.zip>.

A .zip file with skeleton JAVA code is online at:

<http://wwwx.cs.unc.edu/~sjguy/comp575/SquareJava.zip>

OpenGL should already be installed on your computer. It’s a specification which allows your programs to talk to the graphics card. Depending on your language of choice, you will need other libraries as well (see below).

When downloading libraries, make sure you are consistent using the 32bit version of everything (or the 64 bit version of everything). If you try to match 32 and 64 bits, you often get cryptic errors.

To install the needed libraries you generally need to do 3 things.

- 1) Place the library files (.h/.lib or .jar) in a known location
- 2) Tell your development environment where the library files are installed
- 3) Place the library executables (.dll) in a common place (e.g. C:/windows/system32)

C++ Instructions:

You will need GLUT. This is a tool which sets up windows and handles mouse and keyboard inputs.

GLUT can be found here: <http://www.xmission.com/~nate/glut.html>

Java Instructions:

You will need JOGL. These are OpenGL bindings for Java. The solution also uses AWT and Swing, which are windowing libraries that come distributed with Java.

JOGL can be found here (I used version 1.1.1a):

<https://jogl.dev.java.net/servlets/ProjectDocumentList?folderID=11509&expandFolder=11509&folderID=11720>

You will also need to install the gluegen-rt files which are bundled with JOGL.

Resources:

www.google.com <- This is the most helpful of all

<http://www.glprogramming.com/red/> <- The official way to learn OpenGL. Also known as the "Red Book". This is very comprehensive, but it is good to have as a reference.

Hints:

-Once you get the needed libraries set-up, the best next step would be to display a simple stationary square, you will likely have to read up some documentation on this: `glBegin(GL_POLYGON)`, and `glVertex2f(...)` would be particularly helpful to understand

-You should only need to modify `square.h/.cpp` or `square.java`. If find yourself modifying `main.cpp` or `GLViewer.java` your are probably doing something wrong (or implementing extra credit).

-In Java I used `gl.glBegin(GL.GL_POLYGON)` and `gl.glVertex2d(...)` calls (among others). In general you should be able to adapt most C++ examples to Java examples without too much trouble.

-Translating and scaling the square is substantially easier than rotating it, don't be too worried about rotation until you understand how to implement everything without rotation. If everything but rotation works very well, you'll get the vast majority of the credit; if nothing works at all your grade will be significantly lower.

-If you need to compute the arctangent, use the `atan2()` function, it's better than `atan()`.