

CS590G: Assignment #0 – Warm-up

Out: August 31, 2004

Back/Due: September 14, 2004

Objective:

This objective of this assignment is a simple warm-up program to help setup your programming and graphics environment. This assignment will require you to setup a basic PC Windows OpenGL and GLUT program. It is to your benefit to write the program modularly and with a clean setup so as to facilitate subsequent assignments. You have two weeks but it should take you much less time.

Description:

The assignment is to implement a simple image loader and feature clicker. Write a program that given the name of an image file on the command line, provides the user with a window containing the image. The image should cover the entire window. The initial window should be the same size as the image (e.g., 640x480). The window should be resizable using the mouse and standard Windows resizing (upon resizing, the image will need to be redrawn and scaled to fit the whole window). The user should be able to click on an image feature (i.e. a point) and obtain the image coordinates of that feature. The coordinates may be reported using `printf`'s in the console window or in the main window using OpenGL/GLUT text. Note: even if you stretch the window, the printed window coordinates should be in the range of the original image size (e.g., 640x480) – you may optionally round the pixel coordinates to the nearest integer coordinates or display floating point pixel coordinates.

I am providing example images on the course webpage:

<http://www.cs.purdue.edu/~aliaga/cs590g>

The image is in PPM format. This is a simple lossless, uncompressed, image format. You can support other image formats if you like, but you should at least support this one. Included in this handout is a description of the PPM format.

To give in the assignment, email me (aliaga@cs.purdue.edu) your executable by 10:00am BEFORE class on September 14, 2004. It is your responsibility to make sure the email is delivered/dated before 10:00am on September 14, 2004; otherwise it will be considered late. **Hint: don't wait until the last moment to hand in the assignment.**

IMPORTANT: in your email message, please include a short message saying “This is Assignment #0” and attach your executable inside a .ZIP file, do not email a .EXE file as spam filters will probably filter-out the attachment. If you wish me to email you a confirmation of receipt, please indicate so, but I will bring a list of the assignments I have received to class.

For grading, I will run the program with a PPM image on the command line and click on several image features using the mouse. For example:

```
C:\ <your program> pirana.ppm
<window pops up, will click on several features and should get image
coordinates in the format "(x,y)", for instance>
<ctrl-c to end>
```

OpenGL/GLUT/GLUI Primer

If you are fluent in OpenGL/GLUT (/GLUI), this primer is useless. If your OpenGL/GLUT (/GLUI) is rusty, this primer might help refresh your memory. If you are an OpenGL/GLUT whiz, please see me ☺.

OpenGL is a platform independent graphics programming API. An OpenGL program sets up default parameters for rendering and then calls function during program execution to change rendering parameters, draw objects, etc. It comes standard with Windows PCs. More information about OpenGL can be obtained at <http://www.opengl.org> or the book "OpenGL Programming Guide" is a common text format for OpenGL programming.

GLUT is a platform independent windows and event control API. Used together with OpenGL, it allows the user to create windows for graphics drawing as well as handle callbacks for several graphical events (e.g., refresh, resize, move, keyboard input, mouse input, etc). More information about GLUT can be obtained from the website <http://www.xmission.com/~nate/glut.html>.

GLUT is freely available but must be installed. If GLUT is not installed on your PC, you may follow the directions at the above website to install a precompiled DLL on your computer. I will also place a copy of the latest DLL on the course webpage. Basically, you have to copy the DLL into the Windows/System32 folder and the header file into your included path, for example, Visual C/C++ include directory.

GLUI is a platform independent user-interface builder. You do not need to use this library but it can embellish your program very easily. There are other similar libraries out there but I recommend this one for its simplicity yet completeness. Similar to GLUT, you must install this library as well. I will also place installation instructions on the course webpage. More information about GLUI can be found at <http://www.cs.unc.edu/~rademach/glui>.

A basic OpenGL/GLUT (/GLUI) program looks like the following:

```
#include <stdio.h>
#include <GL/gl.h>
#include <GL/glut.h>           // depends on where it is installed...
#include <GL/glui.h>          // if using GLUI...

my_function1()
{
    // my custom function 1
}
```

```

my_function2()
{
    // my custom function 2, etc...
}

redraw()
{
    // OpenGL calls to redraw the window
}

keyboard_input()
{
    // code to handle different key inputs
}

mouse_input()
{
    // code to handle mouse clicks
}

idle()
{
    // code to be executed during idle time
}

myGUI()
{
    // build my user interface
}

main()
{
    // OpenGL initialization (default rendering parameters, etc)
    // GLUT initialization (create a window, setup callbacks, etc)
    // GLUT initialization (call myGUI, etc)

    glutMainLoop(); // enter GLUT main loop
}

```

PPM format

The Portable-Pixel-Map (PPM) format is part of the NETPBM package. PPM format is used for color images, PGM is used for grayscale images, and PNM is used for either. The beginning of a typical PPM file has the following fields:

```

<type>
<image size>
<max color-levels>
<data>

```

The type field can be “P3” or “P6”. P3 is used when the subsequent data is in ASCII format of the form: “R1 G1 B1 R2 G2 B2 R3 G3 B3 ...” where each of R, G, or B is the

value of that particular color for the n'th pixel (pixel order goes left->right, top->down). P6 is used when the subsequent data is in binary format of the same form as above but not in ASCII, rather directly as numbers. This format is more compact but not readable as text.

The image size field is something like "640 480" or "512 512", etc.

The max color-levels is typically "255" for an 8-bit-per-channel color image.

Thus, a sample header is:

```
P6  
640 480  
255  
<data...>
```

and is very simple to read and parse. On the course webpage is an example of a PPM file with both binary and ASCII data. Take a look at them.