$Id: labg-x11-opengl.mm,v 1.207 2019-04-24 17:15:32-07 - - $
PWD: /afs/cats.ucsc.edu/courses/cmps109-wm/Assignments/labg-x11-opengl
URL: http://www2.ucsc.edu/courses/cmps109-wm/:/Assignments/labg-x11-opengl/

This is a non-credit lab, in preparation for an upcoming project which will involve using OpenGL, which is a graphics toolkit. There is nothing to submit or grade. In order to run an X11 application on the server, and have it display on your own workstation, you will need to install an X11 running on your computer.

1. Installing X11

Depending on what personal computer you use, choose one of the following methods of installing X11.

(a) **X11 on Linux.** X11 is the native windowing system used on Linux. You may use any window manager of your choice.

(b) **X11 on OSX.** The Macintosh OSX operating system is based on BSD (the Berkeley Systems Distribution of Unix). Install XQuartz if you do not already have X11 running on your personal workstation.
https://www.xquartz.org/

(c) **X11 on W*nd*z*.** There are some choices : Xming. Cygwin.
http://www.straightrunning.com/XmingNotes/
http://www.geo.mtu.edu/geoschem/docs/putty_install.html
https://www.cygwin.com/

2. Verifying X11

Once X11 has been installed on your computer, verify that it is working. When running an X11 program repomely but displaying on the local workstation, it is necessary to set up X11 forwarding with the **-X** flag of **ssh**, as in :

```
bash-1$ ssh -X username@unix.ucsc.edu
```

One may run an X11 program by typing the command name, usually by backgrounding the process, as in the following. Note the ampersand (**&**).

```
bash-2$ gv &
bash-3$ xev &
bash-4$ xpdf &
```

3. Running OpenGL programs

In the subdirectory **hello-world** there are several OpenGL programs available for testing. If your workstation has X11 properly installed, they can just be run as any other X11 program. Example :

```
bash-5$ glxgears &
```

You are not expected to understand the code in these programs at this time, but you should run the programs to verify that you can. at the end of the line. Three of the programs draw flags on the screen. You can move them around and resize them.

```
bash-6$ bonjour-le-monde &
bash-7$ ciao-mondo &
bash-8$ hallo-welt &
bash-9$ konnichiwa-sekai &
```

The program **hello-world** can also be moved around and resized, and it also reacts to inputs you give it.

```
bash-10$ hello-world &
To cycle the colors: right mouse button or key 'c' or 'C'.
To cycle the fonts: left mouse button or key 'f' or 'F'.
To cycle the greetings: middle mouse button or key 'g' or 'G'.
To invert the colors: key Backspace, Return, or Delete.
To quit: key 'q' or 'Q' or ESC.
```

An X11 window dump of this program is shown at the end of this document (Postscript and PDF versions only).

4. Problems?

Because of incorrect configuration of OpenGL on the server, it is possible that you might see the following message :

```
freeglut (./spincolors):  ERROR:  Internal error <FBConfig with
necessary capabilities not found> in function fgOpenWindow
```

The following statement in your **.bash_profile** might solve that problem :

```
export GLESLIB=/usr/local/android-sdk/emulator/lib64/gles_mesa
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GLESLIB
```

5. Other tests

For other tests of OpenGL, look in the directory **other-tests** and run :

```
bash-11$ glclock &
bash-12$ color-wheel &
bash-13$ teapot &
```

The last one shows a teapot picture both in a solid view and a wire frame. Every time you press the left mouse button the view get swapped.

6. Installing a local OpenGL

Running OpenGL across the internet will be net-intensive, especially if you have a slow internet connection. So you might want to install OpenGL on your own workstation.

http://freeglut.sourceforge.net/

Once you have installed OpenGl on your own workstation, you may copy the files from this **code** subdirectory onto your own workstation and compile them. Depending on the OS you are using, you may have to change the line

```
#include <GL/freeglut.h>
```

to something else, for example :

```
#ifdef __linux__
#include <GL/freeglut.h>
#endif
#ifdef __macosx__
#include <??????????>
#endif
```

The **#ifdef**s detect which operation system you are using and will then include the appropriate header files, thus possibly leading to a portable program. You need to

figure out what the predefined macro identifying your OS, and where the OpenGL header files are.

7. Other references

When looking for documentation on a particular function, look at `https://www.opengl.org/`. For example, to find out about the function `glutPostRedisplay`, type "`OpenGL glutPostRedisplay`" into Google, then select `www.opengl.org`.

Getting Started.
`https://www.opengl.org/wiki/Getting_Started`

OpenGL Programming Guide.
`http://www.glprogramming.com/red/`

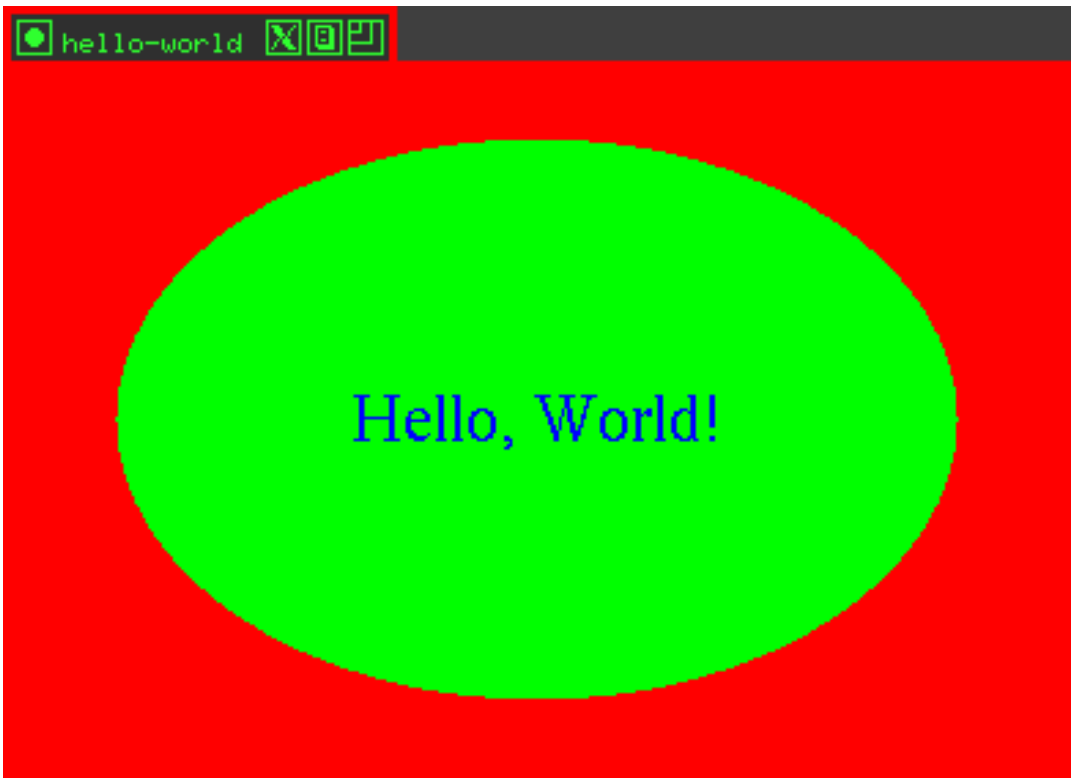
OpenGL Reference Manual.
`http://www.glprogramming.com/blue/`

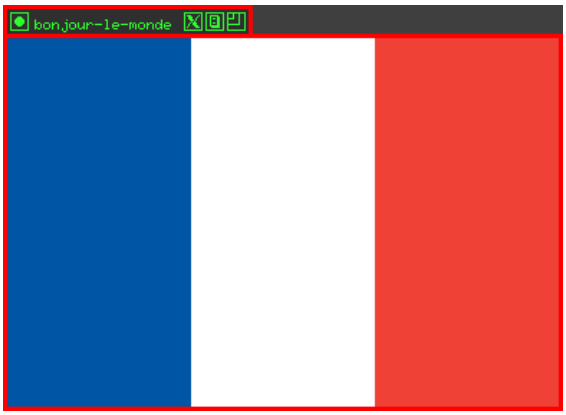
GLUT Tutorial. GLUT stands for the OpenGL Utility Toolkit.
`http://www.lighthouse3d.com/tutorials/glut-tutorial/`

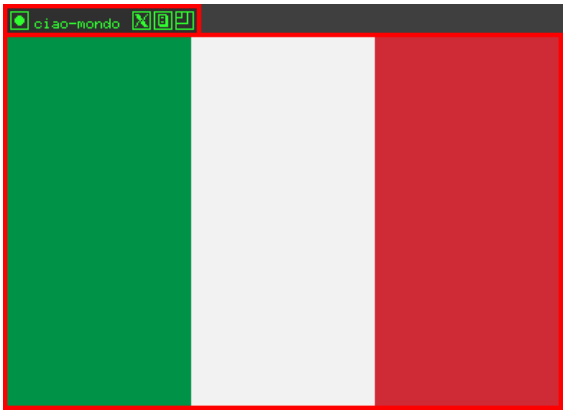
Compiling OpenGL programs on Windows, Linux and OS X.
`http://myweb.wit.edu/wernerm/compilingOpenGl.html`

8. Alternatives

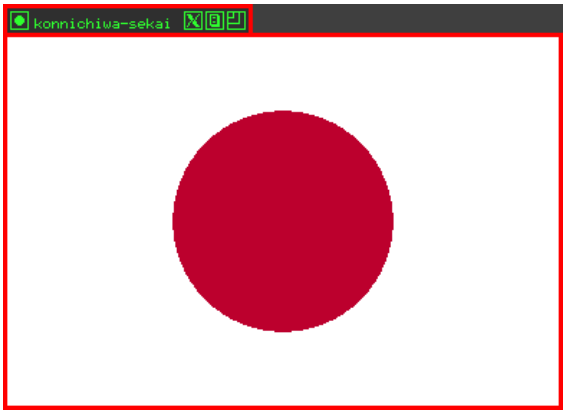
Instead of installing X11 on your workstation, you may do your development in the lab. Instead of installing OpenGL on your workstation, you may work remotely and run all OpenGL programs on the timeshares.




bonjour-le-monde


ciao-mondo


hallo-welt


konnichiwa-sekai