OpenGL in C
Due: April 13, 2020 on myCourses by 23:55

Today's lab will explore creating programs in C that use OpenGL . This lab comes with code that you will need to use. The example code runs on Windows. If you have trouble running the supplied code, there is an alternate Unix activity.

Exercise 1: Run the sample code

"Example.zip" is a Visual C++ project that creates an OpenGL window and displays a 2D scene of a house and a moving sun. The windows of the house are texture mapped, the texture is also provided (in the "/Data" folder).

- Start by double clicking on the EXE file, in Windows, to see it run.
- Now review the code: most of the code is initialization instructions to create an OpenGL window. The actual drawing code is in the DrawGLScene() function.
- Try to compile and run the program:
    - Use VC++ from Visual Studio 2010 to compile and link the project.
        - You need Visual C++ (the free Express version works) and the OpenGL libraries (if they are not included in your version of VC++, you can get them from the Microsoft Platform SDK or for simplicity I included only the relevant files in the second ZIP file: MicrosoftPlatformSDK.zip)
    - Note: If for some reason you have errors compiling/linking the project, here are some things you might try doing:
        - remove glaux.lib from the library dependencies (Menu -> Project -> Properties -> Configuration Properties -> Linker -> Input -> Additional dependencies) add "bmp.cpp" to the source files (right click "Source files" from the solution explorer -> Add -> Existing Item -> Browse for "bmp.cpp")

For Linux users:

Since some of you do not have a Windows computer, these links can be attempted to replace the Windows Lab 9 code:

- This link: http://www.linuxjournal.com/content/introduction-opengl-programming Presents instructions on how to display a Triangle in a Linux OpenGL environment.
- Additional help: https://www.opengl.org/wiki/Programming_OpenGL_in_Linux:_GLX_and_Xlib

Exercise 2: Writing an OpenGL program

It does not matter if you were able to get the Windows or the Linux program to run. Using the program that you were able to get working, do the following: draw a triangle and make it rotate in place three times. Slow the rotation enough for the user to see it happen.

HAVE FUN!

WHAT TO HAND IN

- The code for Exercise 2 and a readme.txt file informing the TA that your lab was completed on a Windows or Linux machine. Make sure to add comments stating your name and ID number.

- You may zip everything into a single file called lab9.zip.

HOW IT WILL BE GRADED

This lab is worth 20 points:

- Exercise 2
    - The submission must show two things:
        - The scene from Exercise 1, and . . . 10 points
        - The rotating triangle from Exercise 2 . . 10 points

Part marks are awarded for incomplete work. The part marks are awarded proportionally to the amount of work you were able to complete successfully.