



FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

BITM 3213 - INTERACTIVE COMPUTER GRAPHICS

LAB SESSION 1

C++ - OpenGL - Setting up Visual Studio

(FROM WEBSITE: <http://www.badprog.com/c-opengl-setting-up-visual-studio>)

C++ - OpenGL - Setting up Visual Studio

Submitted by Mi-K on Sunday, July 26, 2015 - 7:08pm

FROM WEBSITE: <http://www.badprog.com/c-opengl-setting-up-visual-studio>

The **Visual Studio** version for this tutorial will be the **2015 RC**.

The **glew** version will be the **1.12.0**.

The **freeglut** version will be the **3.0.0**.

The glew library stands for **GL Extension Wrangler**.

At the end of this OpenGL tutorial you will be able to display a window with a white square displayed on a black background.

First of all

We are going to use the **32-bit** or the **64-bit** version.

You have the choice, and it will be specified in the tutorial.

Notice that the **version is only based on which platform you want to compile for**.

If you plan to create a program to be executed on a 32-bit platform, so use the 32-bit version.

Every platform accept 32-version, so the 64-bit version is only for specific purpose only.

And so the 32-bit version will work on every Windows system.

glew

You'll need glew headers that could be find on the sourceforge website of the OpenGL Extension Wrangler Library:

- <http://sourceforge.net/projects/glew/files/glew/>

freeglut

You'll also need the freeglut version for Microsoft Visual Studio:

- <http://www.transmissionzero.co.uk/software/freeglut-devel/>

Click the link inside the part **freeglut 3.0.0 MSVC Package**.

The name of the file you have to download is: **freeglut-MSVC-3.0.0-2.mp.zip**.

Once you have downloaded it, there is a folder named **freeglut** inside.

Extract it and rename it to **freeglut-3.0.0** in order to have exactly the same name used in this tutorial.

Paths and directories for this setup

The general directory for our setup will be:

- **C:\dev**

The directory where projects are:

- **C:\dev\visual-studio-c++**

The directory where libraries are:

- **C:\dev\lib**

Visual Studio

Open it and create an empty project > File > New > Project > Templates > Visual C++ > Empty Project.

Then write:

- Name: **Shapes2D**
- Location: **C:\dev\visual-studio-c++**
- Solution: **Create a new solution**
- Solution name: **BadprogTutorial**

OK > Finish.

Your Shapes2D project has been created into your BadprogTutorial solution.

Let's add a main.cpp > Right click your project > Add > New Item > Visual C++ > C++ File > write the name **main.cpp** > Add.

Setting all configurations

Right click your Shapes2D project > **Properties** > On the top left there is a drop down menu > **Configuration** > Select **All Configurations** (instead of Debug)

Including header files (includes)

Right click your Shapes2D project > **Properties** > **Configuration Properties** > **C/C++** > **General** > **Additional Include Directories** > Click it.

On the right there is a drop down menu, click <**Edit...**>.

A new window has appeared: **Additional Include Directories**.

Click the New Line icon > Click the browse button > Then select the two following folders:

1. **C:\dev\lib\glew-1.12.0\include**
2. **C:\dev\lib\freeglut-3.0.0\include**

Click OK > Apply

Including libraries

Library folders

Right click your Shapes2D project > **Properties** > **Configuration Properties** > **Linker** > **General** > **Additional Library Directories** > Click it.

On the right there is a drop down menu, click <**Edit...**>.

A new window has appeared: **Additional Library Directories**.

Click the New Line icon > Click the browse button > Then select the two following folders:

For the 64-bit version

1. **C:\dev\lib\glew-1.12.0\lib\Release\x64**
2. **C:\dev\lib\freeglut-3.0.0\lib\x64**

For the 32-bit version

1. **C:\dev\lib\glew-1.12.0\lib\Release\Win32**
2. **C:\dev\lib\freeglut-3.0.0\lib**

Click OK > Apply

Library files

Right click your Shapes2D project > **Properties** > **Configuration Properties** > **Linker** > **Input** > **Additional Dependencies** > Click it.

On the right there is a drop down menu, click <**Edit...**>.

A new window has appeared: **Additional Dependencies**.

Click the white area and write:

- **freeglut.lib** (click enter to go the next line)
- **glew32.lib**

Click OK > Apply > OK.

Your Visual Studio IDE is now ready to play with OpenGL!

Testing the setup

To test the setup, let's code the most basic code **HelloWorld!** that you can write with OpenGL.

```
#include <GL\glew.h>

#include <GL\freeglut.h>

/**
 * glVertex2f(float x, float y).
 * The point (0.0, 0.0) represents the middle of the window (not the top left corner).
 * The "2f" suffix means 2 values of float type (x and y).
 */

void displayMe(void) {
    glClear(GL_COLOR_BUFFER_BIT);

    glBegin(GL_POLYGON);

        glVertex2f(0.0, 0.0);           // bottom left
        glVertex2f(0.5, 0.0);         // bottom right
        glVertex2f(0.5, 0.5);         // top right
        glVertex2f(0.0, 0.5);         // top left
    glEnd();
}
```

```

    glEnd();

    glFlush();
}

int main(int argc, char** argv) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT_SINGLE);

    glutInitWindowSize(300, 300);          // window size

    glutInitWindowPosition(500, 500);     // distance from the top-left screen

    glutCreateWindow("BadproG - Hello world :D"); // message displayed on top bar window

    glutDisplayFunc(displayMe);

    glutMainLoop();

    return 0;
}

```

On the Visual Studio's top menu there is a dropdown menu with x86 selected.

Click the black tiny triangle on the right of the x86 and select **Configuration Manager**.

A new window has appeared: **Configuration Manager**.

On the right select x86 in the dropdown menu and click **New**.

A new window has appeared: **New Solution Platform**.

On the right click the the Active solution platform dropdown menu and select:

- Type or select the new platform: **x64**
- Copy settings from: **x86** (or Empty)

Click OK.

You could now close the Configuration Manager and as you can see the platform is now specified as x64.

And you could easily click either x86 or x64 (depending of your libraries version).

freeglut.dll and glew32.dll

Before building it, let's copy (not move!) the .dll of these two libraries:

For the 64-bit version

Copy:

- **C:\dev\lib\glew-1.12.0\bin\Release\x64\glew32.dll** (yes this is the right one)
- and
- **C:\dev\lib\freeglut-3.0.0\bin\x64\freeglut.dll**

To the following directory:

- **C:\dev\visual-studio-c++\BadprogTutorial\x64\Debug**

For the 32-bit version

Copy:

- **C:\dev\lib\glew-1.12.0\bin\Release\Win32\glew32.dll**
- and
- **C:\dev\lib\freeglut-3.0.0\bin\freeglut.dll**

To the following directory:

- **C:\dev\visual-studio-c++\BadprogTutorial\Debug**

Building the project

On the Visual Studio's top menu, select either x86 or x64 and build the project.

If there is no error, right click the Shapes2D project > Debug > Start new instance.

If all is OK, you should see a black window with a white square.

Possible Errors

If you had some errors like these ones, it's because you try to play with a different version from the one specified in the setup configuration.

For example if you have set up the configuration with the **64-bit** version, you have to build it with the **x64** configuration.

Same thing for the **32-bit** version, set up with the 32-bit library version and build it as a **x86** configuration.

So double check the paths you've just entered in the setup configuration.

Here the errors:

All packages are already installed and there is nothing to restore.

NuGet package restore finished.

1>----- Build started: Project: Shapes2D, Configuration: Debug Win32 -----

1>main.obj : error LNK2019: unresolved external symbol __imp__glutInitWindowPosition@8 referenced in function _main

1>main.obj : error LNK2019: unresolved external symbol __imp__glutInitWindowSize@8 referenced in function _main

1>main.obj : error LNK2019: unresolved external symbol __imp__glutInitDisplayMode@4 referenced in function _main

1>main.obj : error LNK2019: unresolved external symbol __imp__glutMainLoop@0 referenced in function _main

1>main.obj : error LNK2019: unresolved external symbol __imp__glutDisplayFunc@4 referenced in function _main

1>main.obj : error LNK2019: unresolved external symbol __imp____glutInitWithExit@12 referenced in function _glutInit_ATEXIT_HACK@8

1>main.obj : error LNK2019: unresolved external symbol __imp____glutCreateWindowWithExit@8 referenced in function _glutCreateWindow_ATEXIT_HACK@4

1>C:\dev\lib\freeglut-3.0.0\lib\x64\freeglut.lib : warning LNK4272: **library machine type 'x64' conflicts with target machine type 'X86'**

1>C:\dev\visual-studio-c++\BadprogTutorial\Debug\Shapes2D.exe : fatal error LNK1120: 7 unresolved externals

===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====

Conclusion

Now that your Visual Studio has been set up, you could start creating OpenGL applications.

Well done, you've made it. 🤓