Setting-up OpenGL Dev on Your Computer for Assignments

1. Prerequisites

OpenGL comes pre-installed with every modern graphics card. Make sure you have the latest graphics driver installed.

- AMD/ATI
- Intel
- Nvidia

2. Technical Requirements

You must use the following programming language:

- Microsoft Visual C++

The following libraries have been prebuilt for Windows and are included in the zip file:

- FreeGLUT 3.6.0
  - `win-x64-msvc/include/GL/freeglut*`
  - `win-x64-msvc/bin/freeglut.dll`
  - `win-x64-msvc/lib/freeglut.lib`
- GLEW 2.2.0
  - `win-x64-msvc/include/GL/*glew*`
  - `win-x64-msvc/bin/glew32.dll`
  - `win-x64-msvc/lib/glew32.lib`
- glm 1.0.1
  - `win-x64-msvc/include/glm/*`

The libraries above should be included like:

```
...
#include <GL/freeglut.h>
#include <GL/glew.h>
#include <glm/*>
...
```

As long as you meet the requirements mentioned above, it does not matter how you develop your assignment. You can use any IDE you want, you can use MinGW/MSVC/Clang to compile sources, and you can test/debug sources on a machine of your taste.

However, the TA will assume that your sources compile and run under these conditions. For example, if you managed to include GLEW like `#include <../../GL/glew.h>`, the TA's computer will fail to locate header files and result in a compilation error, which results in zero credits. See section 3. for working set-up examples.

3. Working Set-up Examples

Visual Studio 2022

Download the latest version of Visual Studio here.

1. Navigate to `testbeds/vs` directory provided in the zip file.
2. Open `testbed.sln` and hit `F5`.
3. If you see a blue triangle, you are all set.

MSVC Compiler

MSVC Compilers are Windows-specific C compiler, which is basically shipped with Visual Studio. To use this setting, you should download Visual Studio or their redistributable siblings. This technique is useful when you work with IDEs other than Visual Studio.

1. Open *Powershell*. (can also open by hitting `Win + R` then typing `powershell`)
2. Navigate to `testbeds/msvc` directory provided in the zip file.
3. Run `& .\build.ps1`.
4. If you see a blue triangle, you are all set.

4. Submitting Your Program

You must submit only the source code files (`*.h` and `*.cpp`), excluding IDE-specific files (e.g., `*.vcxproj`, `*.sln`, `.vscode/**`, etc.) as well as the provided libraries (e.g., `*.lib`, `*.dll`). While you may use libraries other than those provided, you must describe the details in your assignment report so that the TA can correctly build your program. These additional libraries must not violate the requirements above.

5. FAQ

What is FreeGLUT?

OpenGL itself is a platform-agnostic specification, and it does not solve platform-specific engineering problems. FreeGLUT helps create an OpenGL context, which would otherwise require platform-specific handling on each platform. There are alternatives like SDL and GLFW, but we chose to use FreeGLUT in this course.

Why do we need GLEW?

OpenGL supports a variety of APIs, and they differ from version to version. Because of that, when we want to use OpenGL, we need to load functions dynamically at runtime. GLEW helps load functions dynamically, a task that would be cumbersome if done manually. While there are similar libraries like gl3w or glad, we chose to use GLEW in this course.

What is the purpose of glm?

glm is a C++ library that provides graphics mathematics, including vector/matrix arithmetic and camera transforms. glm provides an interface similar to GLSL, which makes it easier for OpenGL users. You may also implement a mathematics library on your own.

If you have any further questions, please post your question on Q&A board on PLMS.