Lab 1.1: Assembling the project template

The purpose of this lab is to get comfortable with the process of creating a visual studio project, and incorporating external libraries into your application. Pay careful attention to the suggested directory structures and library version numbers mentioned below.

- Create a new project in Visual Studio 2017.
    - Windows Desktop Wizard
    - Name the project “Template”
    - Application settings
        - Console application
        - Empty project
        - NO precompiled headers
        - NO SDL checks
        - NO ATL, NO MFC

- Open a File Explorer window in the directory Template.vcxproj is located (this is called the **project directory**)
    - Place the source code, shaders and data for the project in the project directory
    - Create a directory called “include” and a directory called “lib”
        - Inside of “include” create a directory called “GL”

- In Visual Studio open the project properties by right clicking on the project in Solution Explorer and selecting Properties
    - VC++ Directories
        - Include directories (configuration: all configurations): add $(ProjectDir)\include
        - Library directories (configuration: all configurations): add $(ProjectDir)\lib

- Glm: This “library” is header only. Copy the inner glm directory into include.
    - In project properties “C/C++->Preprocessor” add a Preprocessor Definition for GLM_ENABLE_EXPERIMENTAL

- Freeglut: Follow the links for Prepackaged Releases for MSVC.
    - Copy lib contents to lib
    - Copy include contents to include
    - Copy bin contents to the project dir.

- Glew: Get windows 32 bit binaries
    - Copy lib/Release/Win32/glew32.lib to lib
    - Copy include to include
    - Copy bin/Release/Win32 to project dir

- FreeImage: Download Windows Binaries
    - Find the directory with the Win32 dll, header and lib files. Put them where they belong

- Assimp: Get 3.0 version with windows binaries
    - Find the win32 Release dll, put it in the project dir
    - Find the win32 Release lib and put it in lib
    - Copy the contents of include to include

- Imgui: This is source only, no libs. From the main page pick Clone or Download-> Download zip
    - Copy the imgui-master contents to your project directory

- Ffmpeg: There are no prebuilt Windows libs, so use the ones from the zip file on Blackboard
    - Keep the headers in their subdirectories, and put the subdirectories in include
    - Put the lib and def files in the lib directory
    - Put the dlls in the project dir

- In Visual Studio Solution Explorer
    - Add the .cpp and .h files in your project directory to the project. Right click Source Files->Add->Existing Item
    - Go back to project properties
        - Add to "Linker-> Input" all of the lib files in your lib directory (configuration: all configurations)

- Build the project and run
    - Be sure you can run both Release and Debug Win 32 builds.
    - Test that you can record a video. It will be saved in the project directory.

- If you get "module unsafe for SAFESEH image" then in Project Properties (Configuration: release) go to "Linker->Advanced->Image Has Safe Exception Handlers" and set to "No".

- Add a filter for shaders: Right-click the project. "Add->New Filter" Rename the filter to shaders. Right-click and add shaders (*.glsl) to the shaders filter.

- Optional: add syntax highlighting for glsl to VS 2017: http://www.horsedrawngames.com/shader-syntax-highlighting-in-visual-studio-2013/
    - Download the vsix file
    - Exit VS 2017
    - Double click vsix and add to VS 2017
    - Even though the blog post describes it as a VS 2013 extension, it still seems to work for 2015 and 2017.