**Resources: setting up the programming environment, compiling and running the demos, troubleshooting**

**Libraries**

We will be using the following libraries for development of software involving computer graphics:

* [OpenGL](https://www.opengl.org/) (core library for 3D graphics, version 4.0 or higher)
* [GLFW](http://www.glfw.org/) (windowing system)
* [GLEW](http://glew.sourceforge.net/) (enables OpenGL extensions)
* [GLM](https://glm.g-truc.net/0.9.8/index.html) (OpenGL Mathematics)
* [SOIL](http://www.lonesock.net/soil.html) (for loading images as textures)

And the following tools:

* [CMAKE](https://cmake.org/)
* Visual Studio 2017 in Windows, or g++ in Linux

**Installing the libraries and tools in Windows**

You can download a package with pre-compiled libraries for Windows systems from [here](https://culearn.carleton.ca/moodle/pluginfile.php/2711709/mod_page/content/19/Libraries.zip). These binaries work only with Visual Studio 2017. Download them and place them in a directory where they can be easily found.  
If you would like to use a different version of Visual Studio, then you need to download the source code of the libraries from the links above, and compile them in your version of Visual Studio. Note that OpenGL itself comes pre-installed in Windows systems. Most of the demos work with OpenGL 3.0, but for a few of the demos, you will need OpenGL 4.0 or higher.  
Also download the latest version of CMake and install it in your system.  
Note also that the libraries and CMake are installed in the game development lab. The libraries are available in the folder "C:\Software\Languages\VS 2017 Libraries".

**Installing the libraries and tools in Linux**

Assuming a Debian-style packaging system, you can install the libraries with the following command:  
*apt-get install libglew-dev libglfw3-dev libglm-dev libsoil-dev*  
You also need to have OpenGL installed (typically a package such as libgl-mesa).  
You can install the tools with:  
*apt-get install cmake*

**Compiling the demos  in Windows**

* Download and uncompress source code of the demo to be compiled
* Provide path of source code to cmake ("Where is the source code")
* Request cmake to build the binaries in the folder <source\_of\_demo>/bin ("Where to build the binaries")
* Confirm that cmake should create the "bin" directory
* Press "Configure"
* Select Visual Studio 15 2017
* If you get an error that says that any of the libraries is undefined, you can ignore that, since we are going to specify the library path next
* Specify the variable "LIBRARY\_PATH" with the path of the downloaded libraries (as explained above)
* Press "Configure" again, and then press "Generate", which generates the visual studio solution file
* Press "Open Project" to open the solution file in Visual Studio
* Select Demo as Startup project
* Build and run

**Compiling the demos in Linux**

* Download and uncompress source code of the demo to be compiled
* Create a bin directory for the binaries, inside the source code's directory: *mkdir bin*
* Enter the new directory: *cd bin*
* Run cmake, telling it to look into the parent directory with the source code: *cmake ..*
* Compile: *make*
* Run the executable generated in the bin folder

**Documentation**

* [OpenGL](https://www.opengl.org/documentation/)
* [GLSL reference](https://www.opengl.org/wiki/OpenGL_Shading_Language) (OpenGL shading language)
* [GLFW](http://www.glfw.org/documentation.html)
* [GLEW](http://glew.sourceforge.net/basic.html)
* [GLM](https://glm.g-truc.net/0.9.8/api/index.html)
* [SOIL](http://www.lonesock.net/soil.html)

**Tutorials**

* OpenGL: [opengl-tutorial](http://www.opengl-tutorial.org/), [open.gl](https://open.gl/introduction)

**Compiling the audio demo in Windows**

* The process explained above should also work for compiling the audio demo. If that does not work, use the following instructions:
* Install the OpenAl library: [Install the core SDK with the Windows installer](https://www.openal.org/downloads/)
* Download and compile the [Alut library](https://github.com/vancegroup/freealut)(remember to set "Release mode" in Visual C++)
* After compiling the Alut library, copy the header files and library to the respective folders in the OpenAl directory. In addition, place the alut dll in the system path
* Set a global environment variable AL\_PATH pointing to the OpenAL folder
* Now it should be possible to compile and run the audio demo
* For extending the simple audio manager, check the references of the OpenAl and Alut libraries:
  + [OpenAl documentation](https://www.openal.org/documentation/)
  + [Alut documentation](http://distro.ibiblio.org/rootlinux/rootlinux-ports/more/freealut/freealut-1.1.0/doc/alut.html)

**Compiling the audio demo in Linux**

* Install the audio libraries: *apt-get install libalut-dev libopenal-dev*
* Compile the source code like the other demos

**Troubleshooting the compilation of the demos in Windows**

* *When running cmake, I get an error about MSBuild.exe not being found.* Recent versions of Visual Studio only come with C# installed but not C++. You need to open the Visual Studio installer and request to add "C++ Desktop Development" to the installation.
* *There is a linking problem when attempting to compile the demo.* Double-check that you are really using Visual Studio 2017, and not 2015 or another version. It can be confusing that VS 2017 is version 15, while 2015 is version 14. The pre-compiled libraries only work with VS 2017.
* *When trying to run the demo, I get an error about ALL\_BUILD not being found.* You forgot to right-click on the demo's project in the solution explorer and select the option "Set as Startup Project".
* *I can run all the demos in Linux, except the particle system demos.* OpenGL 4.0 or higher is not supported by the standard graphics driver in Linux. You need to install the commercial driver specific to your graphics card.