

Visual Computing PS

Computer graphics test code

A. Requirements

1. **C++ compiler:**
 - On **Linux**, you can use GCC, code blocks ...etc.
 - On **Windows**, you should use VisualStudio (community edition is alright).
2. **CMake:** <https://cmake.org/>
3. **OpenGL:**
 - On **Linux**, it should already be provided by the system. However, you can make sure that OpenGL is available by installing the GLUT library: |
`$ sudo apt-get update`
`$ sudo apt-get install freeglut3-dev`
Note: We will not use GLUT (OpenGL Utility Toolkit) in this course, we will use GLFW instead (see below). However, the installation of GLUT contains OpenGL.
 - On **Windows**, the OpenGL library is already contained in the VisualStudio SDK, so there is nothing else to install.
4. Other libraries (GLFW and GLAD) are provided as external dependencies in the **testgl/external** folder, so you don't have to install them.

B. Compilation

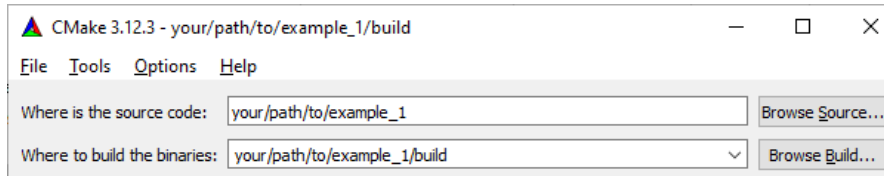
Run CMake. Use the script **CMakeLists.txt** provided to generate a Makefile in an empty **build** folder. Follow the steps below to do so.

Linux:

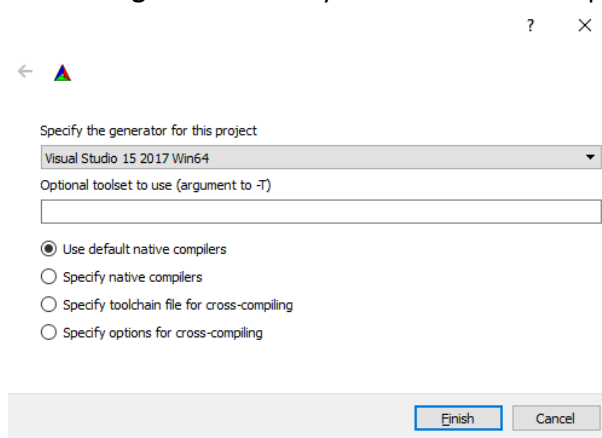
1. Open a terminal in your/path/to/testgl and type:
`$ mkdir build`
`$ cd build`
`$ cmake ../`
Note: with ../ in the cmake command, the path to the directory with the CMakeLists.txt file has to be provided (here it is outside of the build directory).
You can also use the CMake GUI (same as Windows, see below)
2. The folder **build** now contains a Makefile that you can use to compile the code. Just run the following from a terminal inside the **build** folder:
`$ make`
3. Then the executable should be available in the **bin** folder inside of the **build** directory. Open a terminal in that **bin** folder and run:
`$./testgl`

Windows:

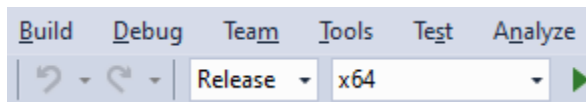
1. You can use the CMake GUI (instead of **example_1** use **testgl** as above):



2. Click **Configure** and select your VisualStudio compiler:



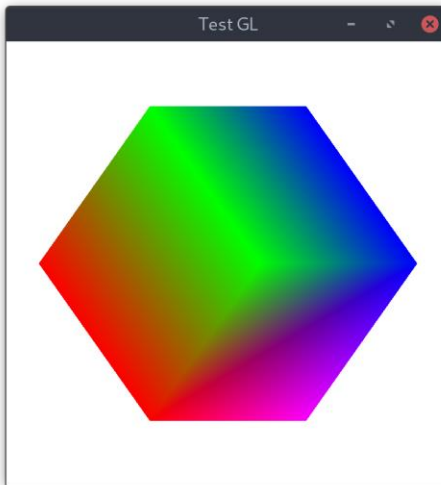
3. When configuring is done, click **Generate**
4. Either click **Open Project** or open the VisualStudio solution (**testgl.sln**) that should now be in the build folder.
5. In the solution explorer (right panel) right click on **testgl_copy_shader** and select build. This target invokes the full build and copies required files (e.g. shaders, assets) into the bin folder.
6. Switch to **Release**



7. Compile the code: **Build -> build solution**
8. Run the program: **Debug -> Start without debugging**

C. Result

You should get a window with one colored cube, as depicted in the figure below:



D. Work from the ZID lab rooms

If you want to work at the University, you should go to room RR21 or RR22. Use Linux to compile and run the code as described above.

E. Troubleshooting

- If you encounter the error message *"Could not open shader file"* when running the program, make sure that you run the program from inside the **bin** directory, that there is a **shader** folder inside of the **bin** directory and that there are the shader files **default.vert** and **default.frag** inside of the **shader** folder. If this is not the case, move to the **bin** directory and copy the **shader** folder in **testgl/src** to the **bin** directory.
- If you experience issues to run the code on your laptop on Linux, try the following command line to run the program:

```
MESA_GL_VERSION_OVERRIDE=3.2 ./testgl
```

Contact: nikolaus.rauch@uibk.ac.at und stefan.spiss@uibk.ac.at