CSGI.GA.2270 – Computer Graphics General Rules for 2022 Fall Assignments

Plagiarism Note and Late Submissions

Copying code (either from other students or from external sources) is strictly prohibited. We will be using automatic anti-plagiarism tools, and any violation of this rule will lead to expulsion from the class. Late submissions will be accepted with a grade penalty of 30% for up to 3 days after the deadline. In case of illness or emergency, please apply for an extension by sending an email to gk2409@nyu.edu attaching relevant documentation.

Provided Software

For this class, you will use the minimal framework provided with each assignment. It compiles on Windows, Linux, and Mac. If you have trouble compiling, follow the procedure used by the auto-builds when available (file `.travis.yml` for Mac/Linux, and file `.appveyor.yml` for Windows) before contacting the assistant. No special hardware is required for this class as we will not be using the GPU for the assignments.

Preparing the Build Environment

The assignments will use CMake as a build system. Before we can begin, you must install CMake on your computer. I recommend installing it with a package manager instead of the CMake download page. E.g. on Debian/Ubuntu: `sudo apt-get install cmake`, with HomeBrew on Mac:`brew install cmake`, and on Windows withChocolatey:`choco install -y cmake`.

You must install a C++ compiler: `gcc/clang` on Linux, `clang` on Mac, Visual Studio on Windows. If you are looking for an IDE to develop in C++, we recommend Visual Studio Code on Mac/Linux, and Visual Studio on Windows.

Compiling the Sample Projects

We will provide a folder for each assignment with some sample code to get you started. Included in each assignment folder is a CMake build system that should enable you to compile your code easily. For each assignment, you will need to do the following:

- Download the assignment code from Brightspace
- Create a directory called `build` in the assignment directory, e.g.:
- `3. cd assignment_X; mkdir build`
- 4. Use CMake to generate the Makefile/project files needed for compilation inside the `build/` directory:
- `5. cd build; cmake –DCMAKE_BUILD_TYPE=Debug ..`
- 6. Compile and run the compiled executable by typing:
- `7. make; ./assignmentX`
- If you run into problems, please reach our CA.

What to Hand In

The delivery of the exercises is done using Brightspace. The repository should follow the template provided in the starter code, and it must contain:

- A short report (in pdf or text/markdown format) containing a description of what you've implemented, as well as explanations/comments on your results. The report must include the name and version of both your operating system and of the compiler that you used.
- The source code, together with the necessary CMake project files, but excluding all compiled binaries/libraries. Specifically, do not include the `build/` directory. The code must successfully compile on your operating system of choice: code that does not compile will not be awarded any point, even if it contains a partial implementation of the solution.
- Screenshots of all your results with associated descriptions in the report file.
- 1-2 minute video showing your demo (e.g. via a screen recorder)

Note: It will not be necessary to use additional external software for your assignments. If you find that you need to use additional binaries / external libraries other than those provided by us, please first get approval by sending an email to me and the course assistants.

Grading

According to your operating systems you use, the assistants might ask you to join a short zoom grading session to help with potential compilation/runtime issues.

Your submission will be graded according to the quality of your image results, and the correctness of the implemented algorithms. The submitted code must reproduce exactly the images included in your readme.

References: Daniele Panazzo