NYU | TANDON SCHOOL OF ENGINEERING

General Rules

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 generally not be accepted. In case of serious illness or emergency please notify the assistants and provide a relevant medical certificate.

Provided Software

For this class, you will use the minimal framework available at https://github.com/jccosta/CS-GY-6533. It compiles on Windows, Linux and MacOSX. If you have trouble compiling follow the procedure used by the auto-builds (Mac/Linux and Windows) before contacting the assistant. Make sure to pull the more recent changes from the git repository as we might be updating the source code from time to time.

Compiling the Sample Projects

In order to compile the provided project files for a given assignment on your machine, you need to do the following:

1. Install CMAKE

2. Clone the repository with the –recursive option. NOTE: The recursive option is very important and it will not work without it.

3. Create a directory called `build` in the assignment directory `TOPDIR/Assignment_X`, e.g. by typing in a terminal window: `cd TOPDIR/Assignment_X; mkdir build`

4. Create the necessary makefiles for compilation and place them inside the `build/` directory, using the CMAKE GUI (windows), or typing: `cd build; cmake ../`

5. Compile and run the compiled executable by typing: `make; ./AssignmentX`

If you run into problems, please create an issue on the github repository .

What to Hand In

The delivery of the exercises is done using github classroom. The repository should follow the template provided, and it must contain:

- 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 in TRAVIS on the Linux operating system.

- A README file (in pdf format) containing a description of what you've implemented and compilation instructions, as well as explanations/comments on your results.

- Screenshots of all your results with associated descriptions in the README file.

Note: It will generally 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 filing an issue.

Grading

The code will be evaluated on Linux. In order to receive a grade, your code **must** compile on TRAVIS. If it does not pass the auto-build on github, the exercise will receive a grade of 0%.

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.

To ensure fairness of your grade, you will be asked to briefly present your work to the teaching assistant. Each student will have 3-4 minutes to demo their submission and explain in some detail what has been implemented, report potential problems and how they tried to solve them, and point the assistants to the code locations where the various key points of the assignments have been implemented. If you cannot make it to the demo session, please schedule a separate meeting with the assistant in the week after the demo session.