Robotics Engineering course

University of Genoa

SOFA framework installation procedure

Additional plugins: SofaPython3 and Geomagic.

Operating System: Windows

Student Chiara Saporetti **Month:** April 2021

1 Sofa installation on windows

Three things before starting:
1. Sofa is a continuously changing software, which means that this tutorial is only tested for the 20.12.02 version.
2. This pdf is meant as an additional help to the already existing tutorial on the SOFA website, since sometimes this tutorial is not totally up to date.
3. You can download everything you need as you go through the tutorial, or download everything from my files for installation. Good luck!

Sofa is available either in binary form (.exe file) or to be built from sources. At the moment, the Geomagic plugin is available only in the sources version, so it's mandatory to choose the building from sources option.
Note: I used a **Windows architecture x64**, using **Visual studio 2019** (but you can use any version of VS $\geq$ 2017), Git (optional) and **pip**.

1.1 Dependencies

- **Microsoft Visual Studio**. Version $\geq$ 2017 from VS 2019.
  In the installer, you must enable:
  In the main panel: the C++ development toolkit, called C++ Build Tools or Desktop C++.
  In the side panel: the C++ ATL and C++ MFC components.
  If you already have VS and want to understand if you have those features: Tools: Get Tools and Features.

- **CMake**. Version $\geq$ 3.12 from CMake: Windows x64 msi and choose the option shown in 1. Check if the variable has been added to your path by going to your computer's Advanced system settings $\longrightarrow$ Ambient variables $\longrightarrow$ System variables. Click on path and look if the Cmake installation folder is there. Should be something like: /path/to/your/CMake/bin. If not: enter folder/bin. Copy path. Then go to your computer's Advanced system settings $\longrightarrow$ Ambient variables $\longrightarrow$ System variables. Click on path and then on Modify. Add as last path line the path to the bin folder you previously copied. Click ok. Check if the installation worked by opening a terminal and typing cmake.

  If you want to download the Zip instead: move the zip file to a desired position and unzip it. Then add cmake to your system varibles.

- **Qt**: Version: $\geq$ 5.12.0 from Qt. Choose Custom installation and then tick the elements specified in Fig. 2

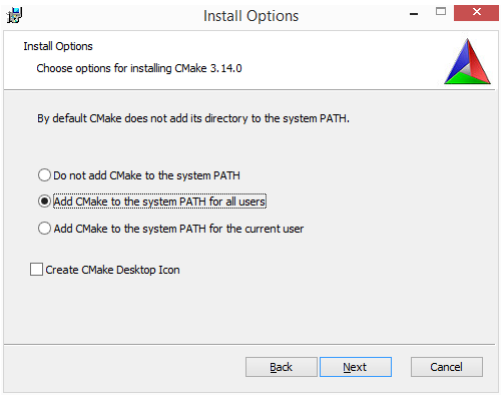

Figure 1: cmake installation

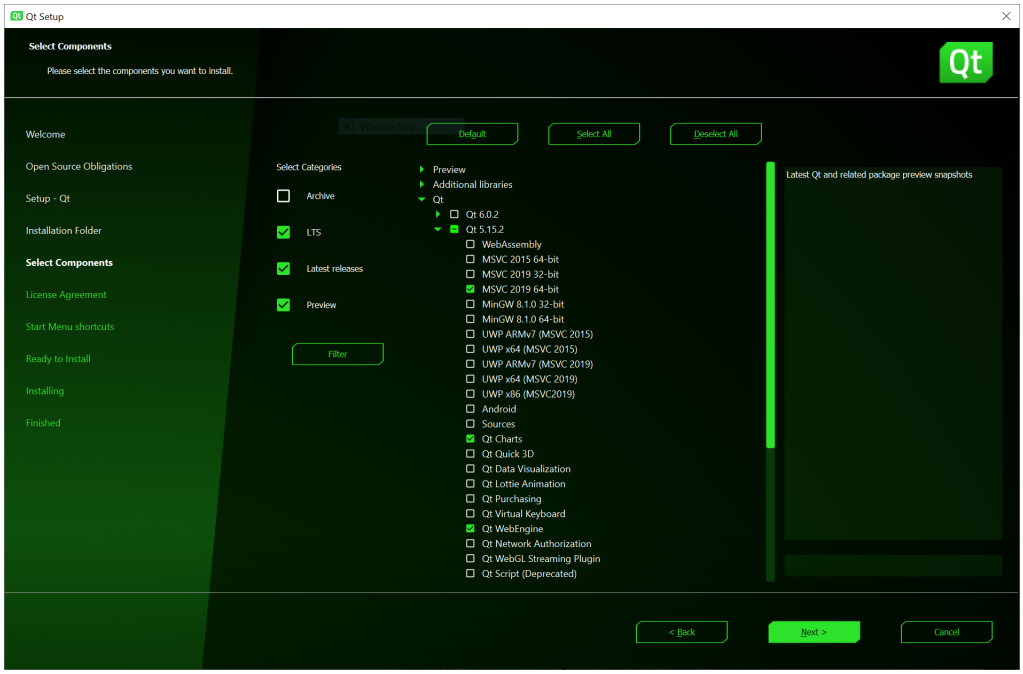

Figure 2: Qt elements needed

- **Boost**. Version $\geq$ 1.65 from Boost: Choose a version and click it (Note: Boost 1_71 is the newest version with python 3.7, so that is the one I suggest to download). Users with Windows 64-bit and VS 2019: choose boost_X_X_X-msvc-14.2-64.exe.

- **python 3.7.x**: I downloaded the exe file 3.7.9 from Download Python3.7 the Windows x86-64 executable installer and then followed instructions.

- **Eigen** from Download Eigen. Version $\geq$ 3.2.10 (I downloaded the latest stable release which is 3.3.9). Unzip to desired location.
- Install **other libraries**. Download from: Download WinDepPAck.
- Add paths to the ambient variables:
  C:_1_71_0
  C:/boost_1_71_0 lib64-msvc-14.2
  C:/Qt/5.15.2/msvc2019_64 bin
  C:/Qt/5.15.2/msvc2019_64 lib
  C:/eigen-3.3.9

1.2 Install SOFA

- **Install SOFA**. Open command prompt and go to a desired location for sofa. I suggest to put it in C. Then type:
  mkdir sofa
  cd sofa
  mkdir src
  cd src
  git clone https://github.com/sofa-framework/sofa.
  Then unzip the previously downloaded other libraries in the same folder sofa/src.
- **Build SOFA**. Inside your sofa folder create other folders so that you have the structure in Fig. 3.

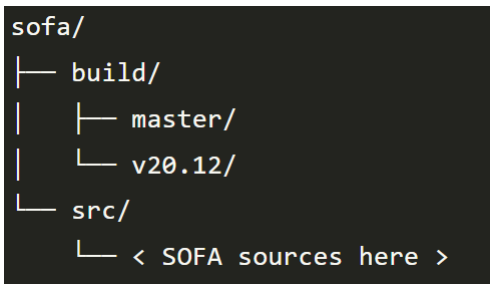

Figure 3: SOFA folder structure

In Windows Start menu, search for Native Tools Command Prompt and run the one correponding to your Windows architecture (x64 for 64-bit) and run:
cmake-gui

If you get the error 'cmake-gui' is not recognized as an internal or external command, it means that your system PATH does not correctly include the path to cmake-gui. In this case, you need to provide the full path to your cmake-gui.

In CMake-GUI, set source folder as sofa/src and build folder as sofa/build/v20.12. Then run Configure. A popup will ask you to specify the generator for the project. Select Visual Studio 16 2019 Win64.

If you have an error with Eigen not found: set, from the cmake gui, the EIGEN3_INCLUDE_DIR to the folder in which you have eigen.

If you have an error with Qt not found: click on Add Entry and add CMAKE_PREFIX_PATH with path to your Qt directory (navigate until msvcXXXX_XX directory).
Example: CMAKE_PREFIX_PATH=C:/dev/Qt/5.11.3/msvc2017_64. Then click Configure and then Generate again.

When you are ready, run Generate. In the build directory, this will create a Visual Studio project (Sofa.sln).

- **Compile SOFA**. From CMake gui click Open Project (or look for runSofa.sln in build folder and click it). Select release option as in Fig 4). Click Build then

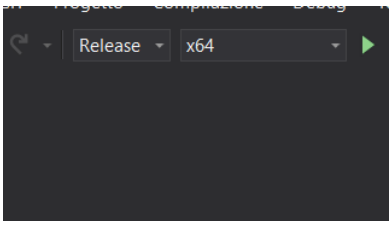

Figure 4: Visual studio parameter

Build solution. (It akes a long time to build).

- To run sofa: on the command prompt:
cd your/path/to/sofa/build/bin/Release
runSofa.
It may take a while the first time!

1.3 Install Python3 plugin

Requirements: Sofa, Python3 (already installed), pybind11: *pip install pybind11.*
In your environment variables modify Path variables by adding:

- sofa install folder. In my case: C:/sofa/build/v20.12/bin/Release.
- python install folder. To understand where Python is installed: In your Python interpreter, type the following commands:
  import os
  import sys
  os.path.dirname(sys.executable).
  It should be something like:
  C/Users/YourName/AppData/Local/Programs/Python/Python37.
- pybind11 install folder. It should be something like:
  path/to/python/Python37/Lib/site-packages/pybind11

Then open cmake-gui. Set variable SOFA_FETCH_SOFAPYTHON3 to true and run Configure.
Then Check variable PLUGIN_SOFAPYTHON3 and run Configure then run Generate. Open the project and build it.
Now, by launching runSofa, you should have the plugin installed (check in edit: plugin manager).

Note: if it is not there: go to bin/Release, and copy-paste the plugin_list.conf.default. Then, in the copied file, modify the extension to plugin_list.conf. Then open it by writing SofaPython3 NO_VERSION on top of the list. Save. Also check if in your build folder there is a python3 folder. If it is, move the python3 folder to build/bin. HOWEVER: I signalled this error to the developers, so it should be corrected in the new version of their documentation.

1.4 Install Geomagic plugin

Uninstall previously installed OpenHaptics SDK or device drivers, if any. Then you must:

- Download libraries: Download OpenHaptics for Windows Developer Edition v3.4 at OpenHaptics libraries. Check that in your environment variables you have OH_SDK_BASE = C:/OpenHaptics/Developer/3.4.0.
- Download device driver at Touch Driver. Plug in the Geomagic Touch (you should see a blue light in the cable). Then plug the USB of the device to your

computer. Run the Touch smart setup. You should see your device on the left after scanning is complete. Configure it by following the instructions.

- To have the plugin: recompile sofa by ticking the PLUGIN_GEOMAGIC variable.

1.5 Build from binaries (no Geomagic plugin)

As I said, the Geomagic plugin is not currently available in the binaries version, but if you don't need it there is a way easier way to use Sofa. To work with SOFA v20.12 + SofaPython3 on Windows, you need to:

- Download and install Microsoft Visual C++ 2019 Redistributable from VS 2019.
- Download and install SOFA v20.12 from SOFA exe
- Download and install Python 3.7 64bit from Python3.7 (Windows x86-64 executable installer).
- Create a system variable SOFA_ROOT and set it to SOFA/install/directory.
- Create a system variable PYTHON_ROOT and set it to Python3/install/directory. Note: to understand where Python is installed: In your Python interpreter, type the following commands:
  import os
  import sys
  os.path.dirname(sys.executable).
- Create a system variable PYTHONPATH and set it to SOFA_ROOT/plugins/SofaPython3/lib/python3/site-packages
- Edit the system variable Path and add at the end:
  PYTHON_ROOT;
  PYTHON_ROOT/DLLs;
  PYTHON_ROOT/Lib;
  SOFA_ROOT/bin;
- Open a Console (cmd.exe) and run
  python -V python -m pip install numpy scipy
- After that, all you need to do is open a Console (cmd.exe) and run runSofa -l SofaPython3 (I think you need to run this command everytime to have SofaPython3)

2 Learning sofa

2.1 Suggestions on how to start

Here are my **personal suggestions** on how to start working with SOFA since I got a bit lost at the beginning.

- Start with watching Hugo's video at [1] until the end of section 'Examples of applications'.
- Then follow SOFA's theory at [2]. (Note: skip tutorial application: it's deprecated.)
- Then go back to Hugo's video and continue it (from User Tutorial part 1).
  Note: check errors both on visual studio (click Visualize then Errors) and on SOFA click Graph.
  From 6:11 it starts with Python code.
- Now that you know something look at the actual SOFA tutorials: go to folder sofasrc examplesTutorials.

Useful link: documentation for classes:
https://sofacomponents.readthedocs.io/en/latest/index.html

References

[1] *Conference on SOFA Framework*. 2020. URL: https://www.youtube.com/watch?v=KHTAgD1oG8Y&t=4307.

[2] *SOFA Tutorials*. URL: https://www.sofa-framework.org/community/doc/using-sofa/runsofa/.