TNM067 — Scientific Visualization
0. Setting up the build system.

September 2, 2022

1 Introduction

This laboratory exercise will help you getting started with the labs and get familiar with the Visualization framework Inviwo. There is no need to present this lab, since the other labs will not be doable without completing this lab. It is highly recommended that you finish this lab before the first lab session.

1.1 Change log

- 2017-08-30: Initial Version
- 2017-09-05: Fix broken links
- 2018-09-04: Fix broken links, Update Source
- 2019-09-09: Update for 2019, Changed suggested paths
- 2020-09-04: Update for 2020, Update for distance teaching
- 2020-09-07: Changed build instructions for Mac
- 2020-10-06: Added more information about lab computers
- 2021-06-21: Added Qt installation instructions
- 2022-06-27: New installation instructions using CMake Presets

2 Documentation

On the following links you can read more about Inviwo and get help with questions related to Inviwo.

- Inviwo: http://www.inviwo.org
- Inviwo API Documentation: https://inviwo.org/inviwo/doc/
- Inviwo Issue tracker: https://github.com/inviwo/inviwo/issues

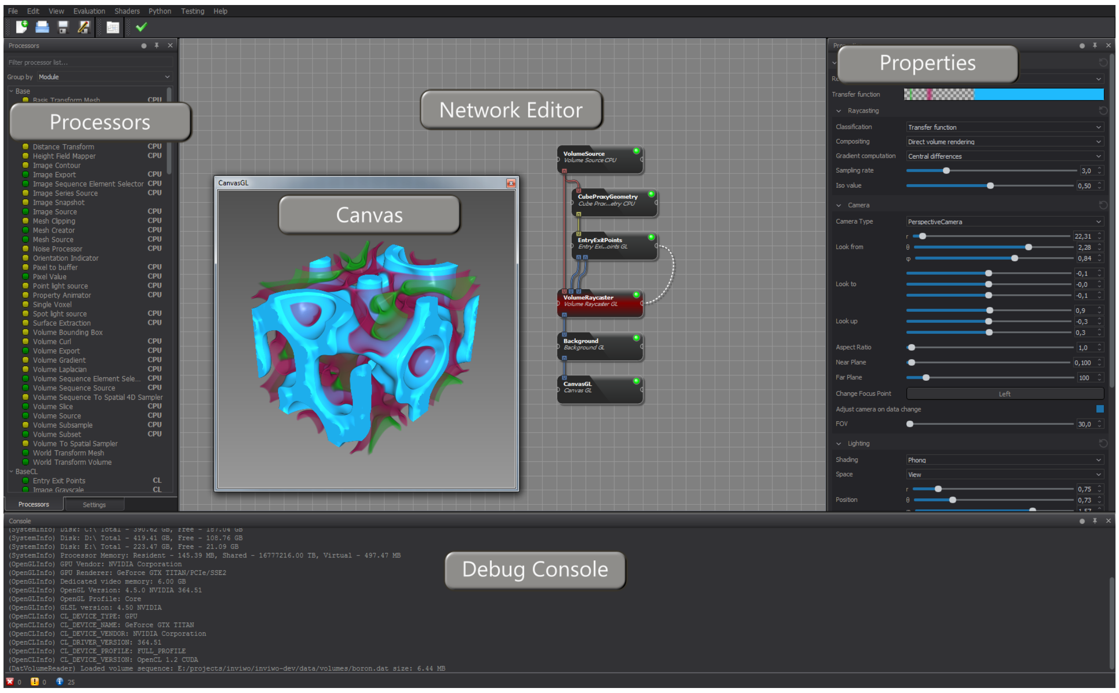
3 The Setup - Get the files, compile and run Inviwo

These labs will be done using the framework Inviwo. Inviwo is a Visualization framework being developed here at Linköping University and which is also in daily use for research. Functional units in Inviwo, called **processors**, are grouped into **modules**. On this GitLab page you can download the set of modules that contains code and stubs for the different classes that you will implement during the lab. The structure for the lab files is shown in figure 2.

To get started with the labs you have to compile Inviwo by following these steps. In the following there are three different sections. One is for setting up Inviwo on the lab computers. The others contain general help for setting up Inviwo on personal machines, Windows and Mac. By following these instruction you should be able to get Inviwo running on you own computer. However, since every machine is different, we can of course not support all personal machines.

3.1 Lab Computers

This section describes how to set up Inviwo on the Windows computers in the lab room. The lab computers are a bit different from your own machines. First, sometimes the path to QT is not put correctly in the corresponding environment variable. Second, it is sometimes hard to find a place to put files, which is not a network resource.



```
(SystemInfo) Disk: C:\ Total - 390.62 GB, Free - 187.04 GB
(SystemInfo) Disk: D:\ Total - 419.41 GB, Free - 108.76 GB
(SystemInfo) Disk: E:\ Total - 223.47 GB, Free - 21.09 GB
(SystemInfo) Processor Memory: Resident - 145.39 MB, Shared - 16777216.00 TB, Virtual - 497.47 MB
(OpenGLInfo) GPU Vendor: NVIDIA Corporation
(OpenGLInfo) GPU Renderer: GeForce GTX TITAN/PCIe/SSE2
(OpenGLInfo) Dedicated video memory: 6.00 GB
(OpenGLInfo) OpenGL Version: 4.5.0 NVIDIA 364.51
(OpenGLInfo) OpenGL Profile: Core
(OpenGLInfo) GLSL version: 4.50 NVIDIA
(OpenCLInfo) CL_DEVICE_TYPE: GPU
(OpenCLInfo) CL_DEVICE_NAME: GeForce GTX TITAN
(OpenCLInfo) CL_DEVICE_VENDOR: NVIDIA Corporation
(OpenCLInfo) CL_DRIVER_VERSION: 364.51
(OpenCLInfo) CL_DEVICE_PROFILE: FULL_PROFILE
(OpenCLInfo) CL_DEVICE_VERSION: OpenCL 1.2 CUDA
(DatVolumeReader) Loaded volume sequence: E:/projects/inviwo/inviwo-dev/data/volumes/boron.dat size: 6.44 MB
```

Figure 1: The initial view of Inviwo, with the most important components annotated.

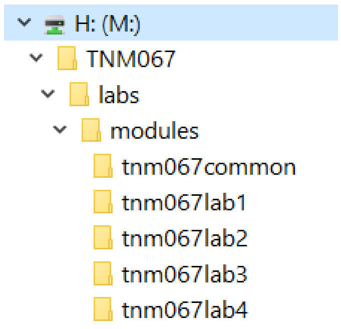


Figure 2: Folder structure for the labs. All modules, when available, should be placed in the **modules** folder or CMake will have trouble finding them.

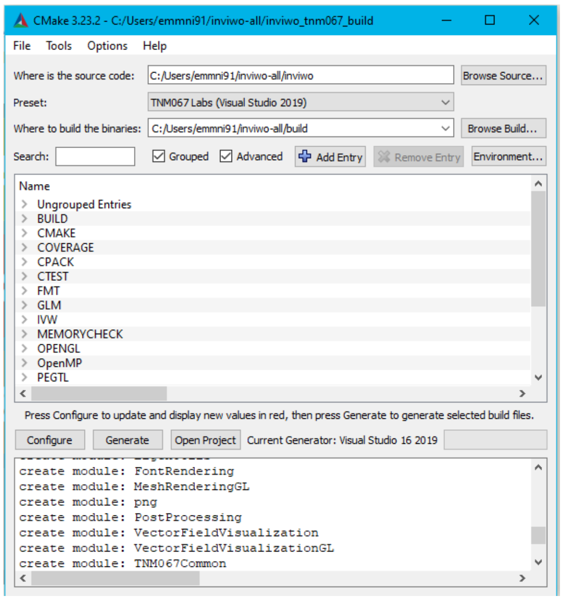
3.1.1 Set-Up

1. Open an Explorer and navigate to "This PC" - "Windows (C:)" - "Users" - "USERNAME"
2. Check that this folder is local on drive C: (**DO NOT USE your "home drive"**)
3. Create a new folder named "TNM067" (See Figure 2)
4. Open Git bash and clone the Inviwo repository including all submodules by running the following git command:

   ```
   git clone --recurse-submodules https://gitlab.liu.se/MIT/tnm067/inviwo
   ```
5. Download / Clone the lab-files, unzip them and put them **next to** "inviwo".

3.1.2 Compile and Run Inviwo

1. Create a build folder for the Inviwo binaries. Should be named something like "build" and put **next to** the folder for the Inviwo source files. At this stage you should have three folders next to each other: **inviwo** containing the source code of the main application, **tnm067-labs-code** containing the modules for the labs 1 - 4 and **build** where you build the binaries.
2. Open the CMake GUI
3. In the CMake GUI, specify the path to the source code of Inviwo: C:\users \USERNAME\TNM067\inviwo and (USERNAME is a placeholder for your username). See another example in figure 3.
4. Check what Visual Studio version is on the computer. In the GUI, beneath the source code path, select a preset matching your Visual Studio version: TNM067 Labs (Visual Studio 2019/2022)
5. Press Add Entry button again. Set the name to `CMAKE_PREFIX_PATH` and change the



CMake 3.23.2 - C:/Users/emmni91/inviwo-all/inviwo_tnm067_build

Where is the source code: C:/Users/emmni91/inviwo-all/inviwo

Preset: TNM067 Labs (Visual Studio 2019)

Where to build the binaries: C:/Users/emmni91/inviwo-all/build

Search:
☑ Grouped ☑ Advanced

Name
- Ungrouped Entries
- BUILD
- CMAKE
- COVERAGE
- CPACK
- CTEST
- FMT
- GLM
- IVW
- MEMORYCHECK
- OPENGL
- OpenMP
- PEGTL

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 16 2019

```
create module: FontRendering
create module: MeshRenderingGL
create module: png
create module: PostProcessing
create module: VectorFieldVisualization
create module: VectorFieldVisualizationGL
create module: TNM067Common
```

Figure 3: An example of how Cmake GUI looks like with specified paths to the inviwo source files and build folder.

type to PATH. Set it to folder where the QT config file is, for example: `C:\Qt\6.2.3\msvc2019_64`.

6. Hit configure once and select the Visual Studio Version you are running on your machine (For example: **"Visual Studio 16 2019"**).
7. Hit Configure one more time and then hit Generate.
8. Open visual studio from within CMake by pressing the Open Project button [1].
9. Change "Solution Configuration" from Debug to RelWithDbgInfo (In the Dropdown menu in the toolbar). This will make the algorithms run significantly faster.
10. From the build menu, select "Build" (You can also press F7 or ctrl+shift+B to build) (Note: This will take sometime).
11. Press F5 to run Inviwo.

[1]Note that once you have generated the solution files using Cmake, you don't have to use Cmake again to open the Visual Studio project. It is enough to just open the Visual Studio solution.

3.2 Own Computers - Windows

This section describes how to set up and build Inviwo on Windows computers. Instructions for developers can be found in the **Inviwo documentation**, but below we provide some more detailed instructions.

3.2.1 Install Dependencies

To be able to compile and run Inviwo you will need to download and install the following dependencies:

- Visual Studio (Version 2019 (> 16.5.7) is necessary, or 2022)
- Qt (at least v.6.2.3) -
  - Scroll down to the end of the page and get the Qt Online installer for binary versions of Qt.
  - Start the installer, which will prompt you to create a Qt account which you now have to do.
  - Use default options until you get to **Select Components**, where you need to pick a Qt version > v6.2.3 Open up that version and select **MSVC 2019/2022 64-bit**. After this you should go with the default options throughout the installation.
  - Add Qt to your **system-wide** variables by creating the variable QT_DIR in the windows environment variables and set it to the path of your Qt installation. Should be something like `Qt/6.2.3/msvc2019_64/bin`.
- Cmake (at least v.3.23.2)
- Git

Make sure to get the Qt version matching your Visual Studio version. We suggest installing Visual Studio before Qt, then the Qt installer will automatically find the version.

3.2.2 Get the Files

1. Create a folder to store the Inviwo installation. Should be on a local drive.
2. Open Git bash and clone the Inviwo repository including all submodules by running the following git command:

   ```
   git clone --recurse-submodules https://gitlab.liu.se/MIT/tnm067/inviwo
   ```
3. Download the lab-files, unzip them and put them **next to** your inviwo folder.

3.2.3 Compile and Run Inviwo using Cmake

The instruction for Windows are the same as for the lab room. 3.1.2 for the other instructions.

3.2.4 Known Issues - Windows

List of possible installation issues we have heard about on Windows.

Python

If you have a Python version installed anywhere on your computer, Inviwo will try to use it for the Python modules (if they are enabled in Cmake). If the Python directory is not in your PATH, you can get an error when trying to run Inviwo. If so, then you have to add the path to the Python directory to your PATH in the list of environment variables. Note that it should be the Python path that Cmake finds during the configuration step (search for python in the Cmake output).

We also recommend to put the Python path on the top of the list, to prevent Inviwo from using any other version (the priority in the list goes from top to bottom).

3.3 Own Computers - Mac

This section describes how to install Inviwo on Mac computers, with focus on how to get the labs to work.

3.3.1 Install Dependencies

- Install Homebrew as package manager
- Download Xcode from the app store
  - Install command line tools with: `xcode-select --install`
- Open a terminal and run the following:
  - `brew install --cask cmake`
  - `brew install qt`
  - `brew install python`
  - `pip3 install numpy`

3.3.2 Get the Files

1. Create a folder to store the Inviwo installation. Should be on a local drive.
2. In the terminal, clone the Inviwo repository including all submodules by running the following git command:

   ```
   git clone --recurse-submodules https://gitlab.liu.se/MIT/tnm067/inviwo
   ```
3. Download the lab-files, unzip them and put them **next to** the inviwo folder.

3.3.3 CMake

1. In the terminal, run:
   `open /Applications/Cmake.app`
2. In the GUI, select *Browse source* and pick the folder where you stored Inviwo.
3. Under Preset, select TNM067 Labs (Unix Makefiles)
4. Click *Configure*, Xcode should be the selected compiler. This first configure pass may take some time.
5. Hit Configure again and then press Generate.
6. Open XCode from CMake by pressing the Open Project button.

3.3.4 Xcode

1. In the pop-up, select *Automatically Create Schemes*.
2. In the upper left-corner, besides the Stop-button, select from the list the scheme *inviwo*.
3. Then, go to the bottom of the list and select *Edit scheme...*, see Figure 4.
4. Under Run/Info, set Build Configuration to *RelWithDebInfo*. The Executable should be set to *Inviwo.app*. It should look like the settings shown in Figure 5.
5. Build Inviwo using the Play-button. This may take a very long time.

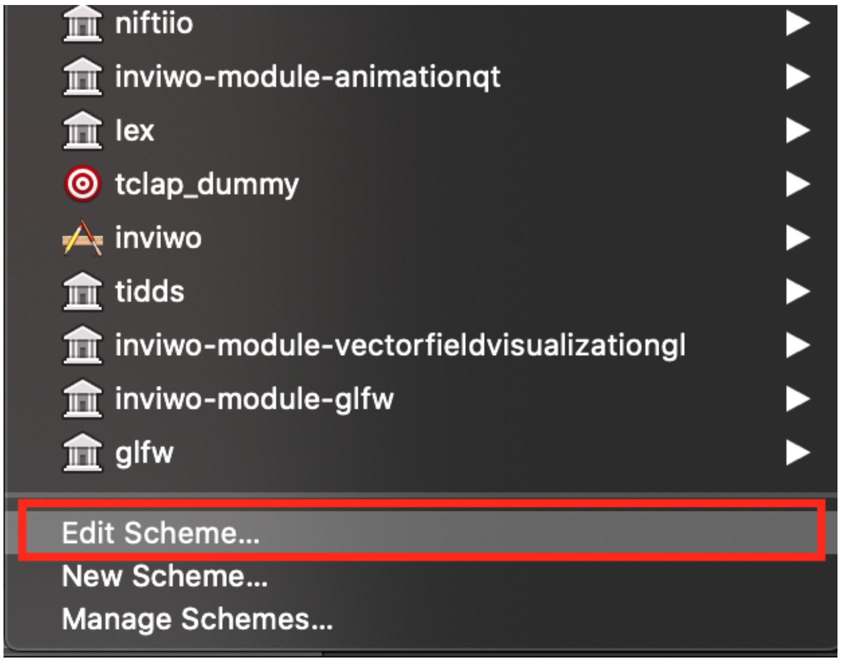
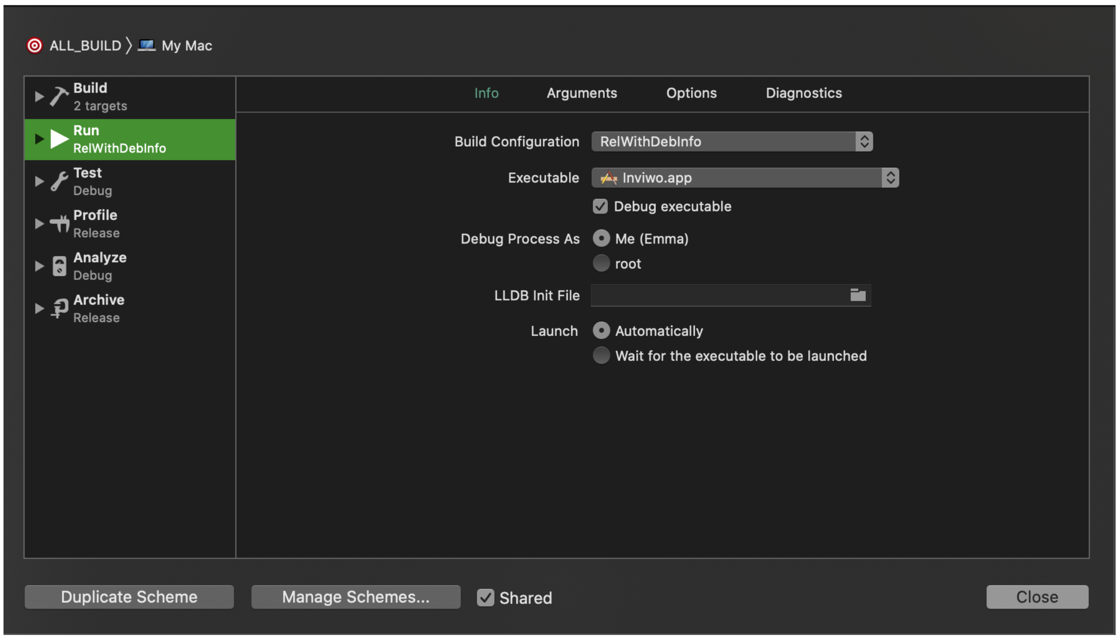


Figure 4: Scroll down the scheme list to the bottom and select Edit Scheme...



Build Configuration: RelWithDebInfo

Executable: Inviwo.app

[✓] Debug executable

Debug Process As: [✓ Me (Emma)] root

LLDB Init File:

Launch: [✓ Automatically] Wait for the executable to be launched

[✓] Shared

Figure 5: What the Run/Info tab should look like with the correct settings.

4 Get used to Inviwo.

Task 1 — Get used to Inviwo:

1. Build and run Inviwo as described above, while it is building, to the next step.
2. Read the http://www.inviwo.org to get overview on the different graphical components in Inviwo and User Manual to get deeper understanding on how they work.
3. Experiment with the different example workspaces (File → Example Workspaces).
4. Get familiar with GUI components such as Processors, Properties, Links, Ports and Port-Connections.
5. Try to create some simple networks, for example loading images from disk and show them in a canvas. Do some image processing such as low-pass filtering.

You should be able show a brief understanding of the following questions:

1. What is a Processor?
2. What is a Property?
3. How does data flow through the network?
4. How does property links work?

You are free to ask questions to the Inviwo team on Github Discussions and to join the Inviwo Slack.