Université de Lille

Lille iut

NTNU

📍 Sverres gate 12, 7012 Trondheim
☎ +47 73 59 95 000
🌐 www.ntnu.no

ERASMUS INTERNSHIP REPORT

Camera Manager – Development of a new software module for integration of Color Cameras

GERMAIN Adrien
June 2019

Supervisors: HOLT Tomas
HARALD NILSEN Jan
HOLT Alexander

IUT A LILLE
Avenue Paul Langevin, 59653
Villeneuve-d'Ascq

Summary

Thanks

First of all, I want to thank my supervisors, Mr. Tomas HOLT, Mr, Alexander HOLT and Mr. Jan HARALD NILSEN who offered me a chance to come here and to work on this project. Also, to have helped us and answers any of our questions.

I also want to thank the IUT of Lille to have trained me during two years for this final project and thank to Mrs. Garau to improve my English and to help me during all my time in Norway and before.

And finally, I want to thank Gauthier LECLERC and Hugo FOURNIER for working with me, be available when I had some question and for all the shared moments together.

Introduction

During my two years degree in electrical engineering I have to make an internship in a private company. But there I could make it in another university in another country, and this is what I have chosen. So I have chosen the NTNU University in Trondheim in Norway. It is an internship of 3 months from the first of April to the thirty of June.

I have made this choice because I wanted to make my English better. But it was not the only thing, it was also to discover a new country. A different way to live, some other traditions that I don't know and new landscapes. I also have chosen the NTNU because it is a good University where the supervisor wanted some good student. The subject were quite interesting as well because I know that they were working on virtual reality and stuff like that which are some subject that I like.

To keep my report structured, I will start with the presentation of the NTNU University before going through my missions and the project, and to end with what this experience brings to me and how the DUT help me success this internship.

The University: NTNU

History

The NTNU (Norges Teknisk-Naturvitenskapelige Universitet or Norwegian University of Science and Technology in English) that we know today is not that old. But it has a quite long history. It begun in the 1700s and precisely in 1760 with the foundation of the first academic society.

One century later, the government wanted to create a Norwegian Polytechnical institute and they founded it in 1870 and was named Trondheim Technical College. But this institute was disbanded in the beginning of 1900s to let his place to the NTH (the Norwegian Institute of Technology) in Trondheim. NTH is the right ancestor of the NTNU of today.

Then on 1 January 1996, the NTNU was created by the merging of six research and higher education institutions which were:

- Norwegian Institute of Technology (NTH)
- Museum of Natural History and Archaeology (VM)
- Norwegian College of General Science (AVH)
- Faculty of Medicine (DMF)
- Trondheim Academy of Fine Art (KiT)
- Trondheim Conservatory of Music

Today

Today, the NTNU is 42 031 students, 395 doctoral degrees and 9.4 Billion NOK (nearly 1 Billion euros) of annual budget. The NTNU is today the biggest University in Norway. It has a large brand of field like science and technology of course, but also humanities, social sciences, economics, medicine, health sciences, educational science, architecture, entrepreneurship, art disciplines and artistic activities.

Localisation

The NTNU has few campus in Norway. They are in the city of Trondheim, Gjøvik and Ålesund.

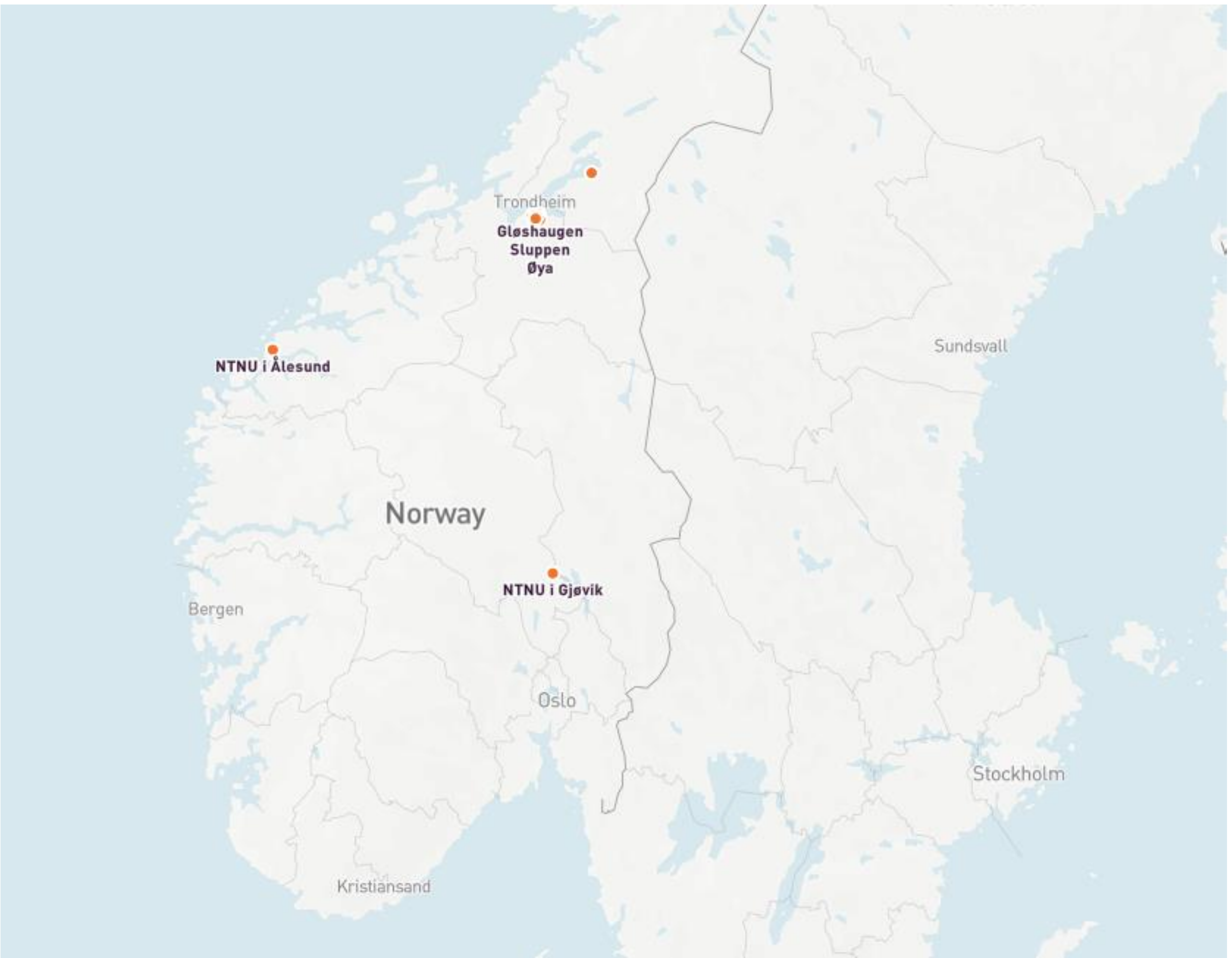


Figure 1: NTNU Campus in Norway

About me, I am in the Trondheim campus. There are a lot of different campus also in Trondheim but I am attached to the IT department and this one is located at Kalviskinnet, in the Trondheim center.

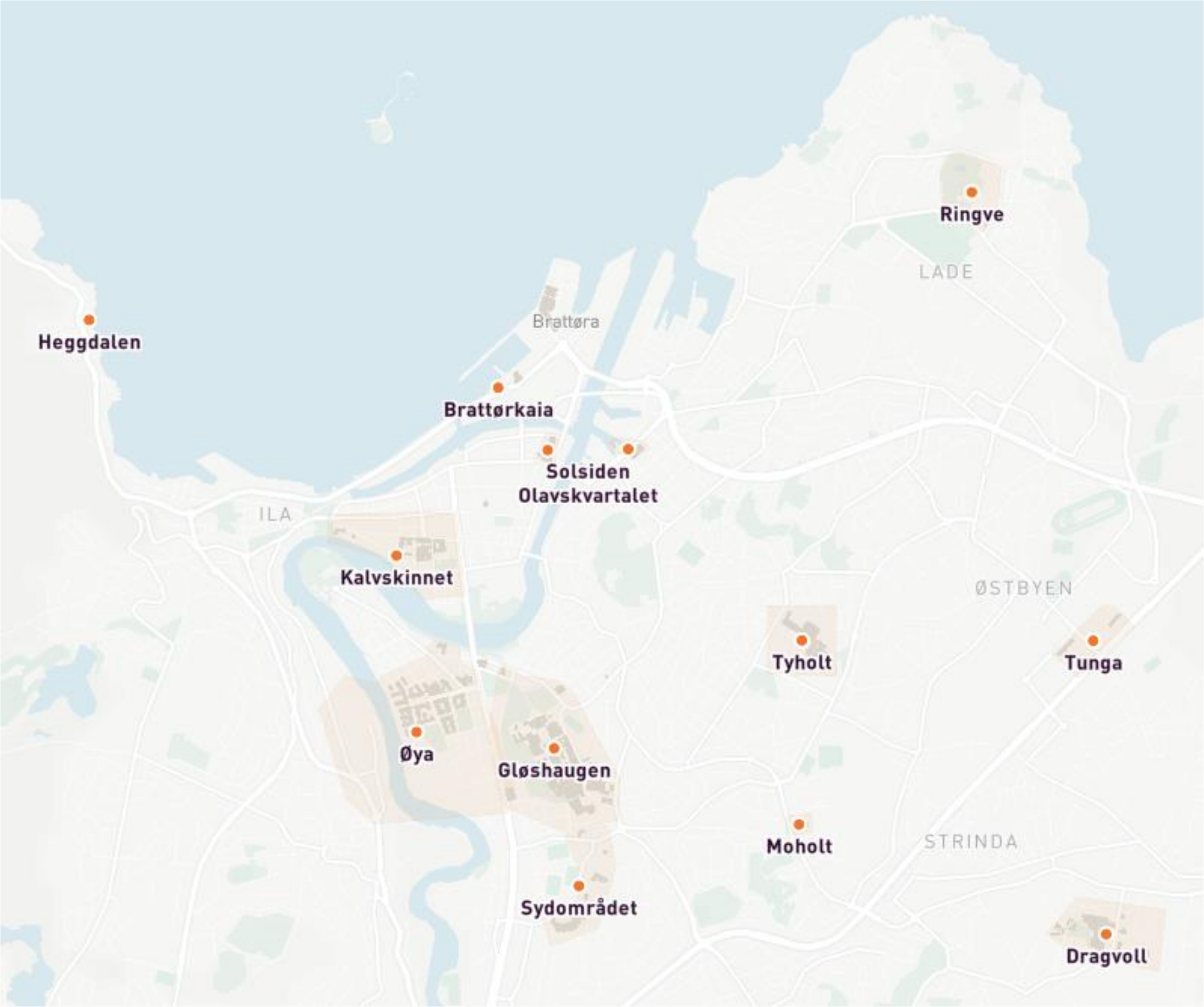


Figure 2: NTNU campus in Trondheim

About my mission

My department

While I am studying electrical engineering, I am attached to the IT department. The housings where the lab is, are at the Kalviskinnet campus in the Trondheim center. My tutors are Jan Harald Nilsen and Tomas Holt and they are both professor in the computer science department at Kalviskinnet. In 2007, they created with some other people a company named 3D Motion Technologies AS which is about virtual reality and motion capture. And this is the company associate with the NTNU we work with.

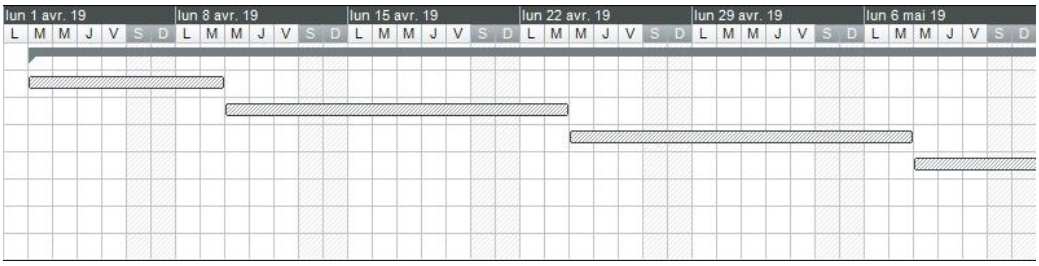
Description of my mission

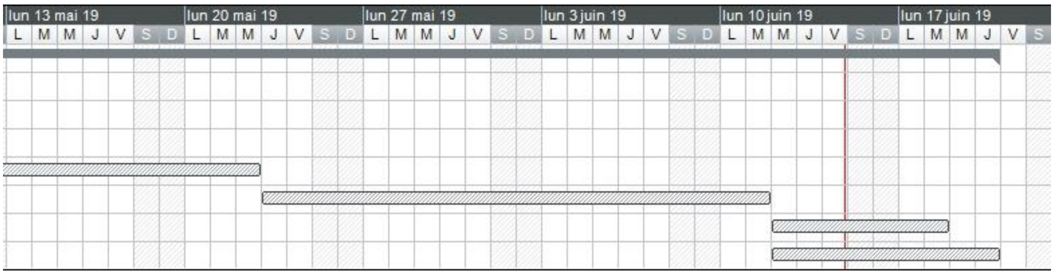
It has begun with the presentation of the laboratory and of what they were doing here. To be able to make motion capture, they have to make a calibration of the cameras. But the calibration is a heavy part and they wanted it to be easier and faster. In that case, they develop an application made in C++ language with Qt Creator which allows them to do what they want. But, this application were developed to work with the FlyCapture API which is the API of the camera they were using. Moreover, now the wants to get some better camera, newer but they no longer work with FlyCapture but with a new API called Spinnaker. Due to that, the application will no longer work as well. So here is our mission, we have to implement the application they were using with the Spinnaker API. Of course it has to work with both FlyCapture and Spinnaker API to continue to use the old cameras.

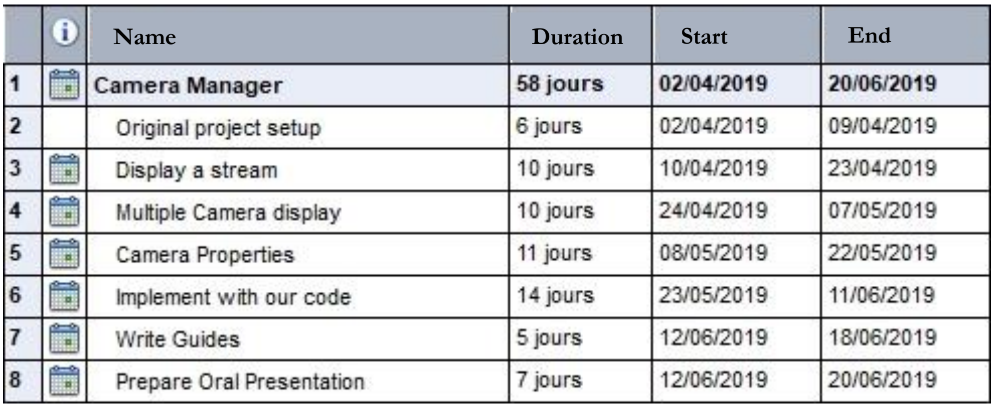
My main mission

Provisional schedule

Here is the provisional schedule we followed. It has been made using the Gantt method and created thanks to the MindView software.



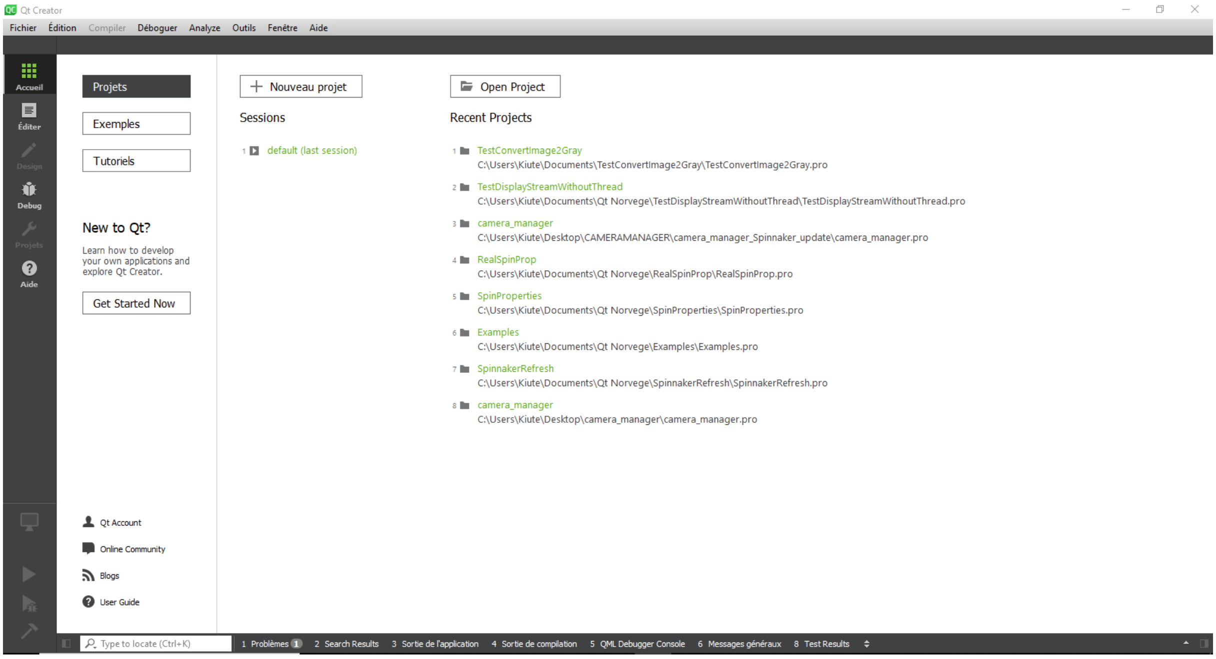




	ⓘ	Name	Duration	Start	End
1	📅	**Camera Manager**	**58 jours**	**02/04/2019**	**20/06/2019**
2		Original project setup	6 jours	02/04/2019	09/04/2019
3	📅	Display a stream	10 jours	10/04/2019	23/04/2019
4	📅	Multiple Camera display	10 jours	24/04/2019	07/05/2019
5	📅	Camera Properties	11 jours	08/05/2019	22/05/2019
6	📅	Implement with our code	14 jours	23/05/2019	11/06/2019
7	📅	Write Guides	5 jours	12/06/2019	18/06/2019
8	📅	Prepare Oral Presentation	7 jours	12/06/2019	20/06/2019

What I have done

First of all, the supervisor just give us the original project and cameras. With this, we had to just make it works on our PC. This step is to configure and set up the working environment by installing Qt creator, the software development kit for the old API Flycapture and all the other things that we need to work properly. I first tried to set up the project on Windows because they shows us the project working on Windows and it were the friendliest operating system for me. So I just started to install the FlyCapture SDK, but with Qt creator, we also needed visual studio. So I decided to use the latest release but it doesn't work. So I've uninstalled all and tried with the release one year before the latest but it hasn't work as well.



1 default (last session)

Recent Projects

1 TestConvertImage2Gray
C:\Users\Kiute\Documents\TestConvertImage2Gray\TestConvertImage2Gray.pro

2 TestDisplayStreamWithoutThread
C:\Users\Kiute\Documents\Qt Norvege\TestDisplayStreamWithoutThread\TestDisplayStreamWithoutThread.pro

3 camera_manager
C:\Users\Kiute\Desktop\CAMERAMANAGER\camera_manager_Spinnaker_update\camera_manager.pro

4 RealSpinProp
C:\Users\Kiute\Documents\Qt Norvege\RealSpinProp\RealSpinProp.pro

5 SpinProperties
C:\Users\Kiute\Documents\Qt Norvege\SpinProperties\SpinProperties.pro

6 Examples
C:\Users\Kiute\Documents\Qt Norvege\Examples\Examples.pro

7 SpinnakerRefresh
C:\Users\Kiute\Documents\Qt Norvege\SpinnakerRefresh\SpinnakerRefresh.pro

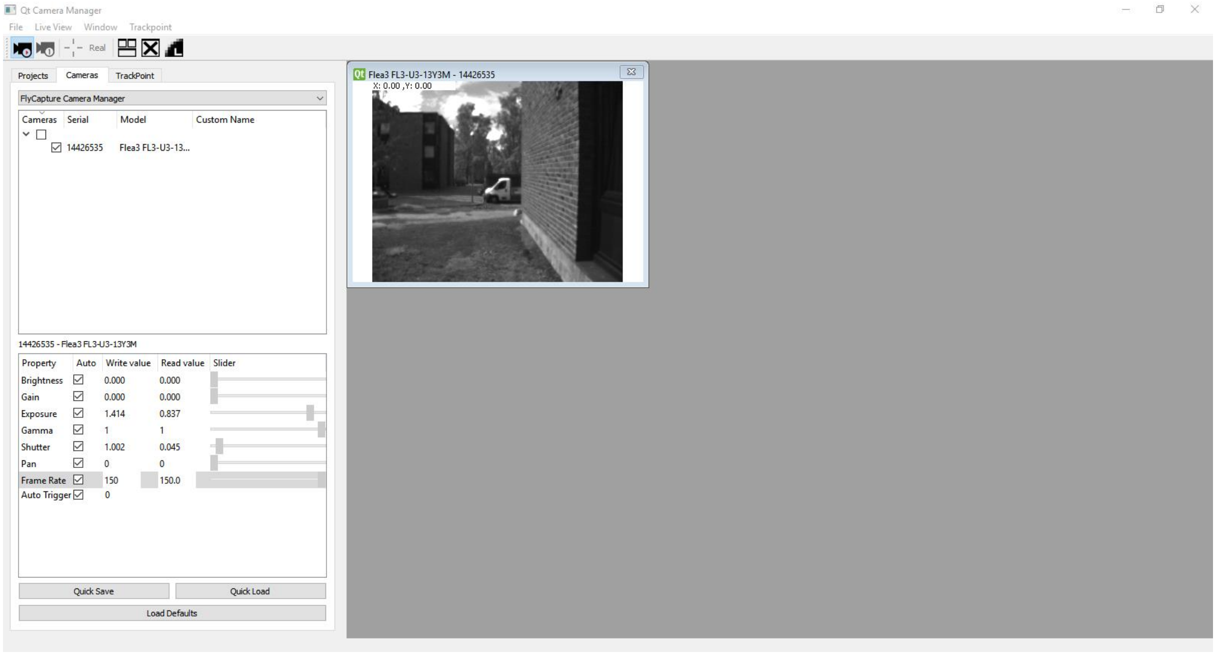
8 camera_manager
C:\Users\Kiute\Desktop\camera_manager\camera_manager.pro

Figure 3 Qt Creator IDE

Due to the very long and heavy installation on visual studio, I have decided to try on Ubuntu to see if it is simpler and if it will work on it. So, I have made the same things that I have done on Windows, I followed the user guide they gave us. I have needed to install the FlyCapture SDK as well, Qt Creator and some libraries like the OpenGL one which is used in the project. After all that things, I have been able to open the project in Qt creator and to configure it with the GCC compiler. But they were some issues during the compilation, and because I have never done C++ before, I was not able to solve them.

Then, I decided to take one week to learn the C++ language. I have used the old website "le site du zero" which is named now "OpenClassroom" which have online courses and exercises to teach a subject which was the C++ in my case. It was not as complicated as I thought because the C++ is an oriented-object programming language, and during my formation in GEII I have worked on java which is also an oriented-object programming language. So the big concept was the same, so I had to learn the distinctiveness of this language and to understand them. A big part of the learning was also to discover and to work with the Qt libraries.

Unless that, I continued to get some issues with the project file of Qt Creator. There is no project file in Java, so I could not try to compare both to understand. But the next meeting was approaching and I really wanted to make the project working on my PC for it. So I have asked the two other student on the project to work together and I could ask them some questions. So I nearly understand how this project file was working but I have not already set up all properly. I continued to work before the meeting to make it work and I finally set up all properly and the project compiled. So after that, I was able to launch the project and to get the video stream of the camera. So it was nearly what I wanted to achieve but I still get some issues like with the camera properties. These issues were solved right after the meeting because the supervisor explain us why (A file was missing inside the project directory).



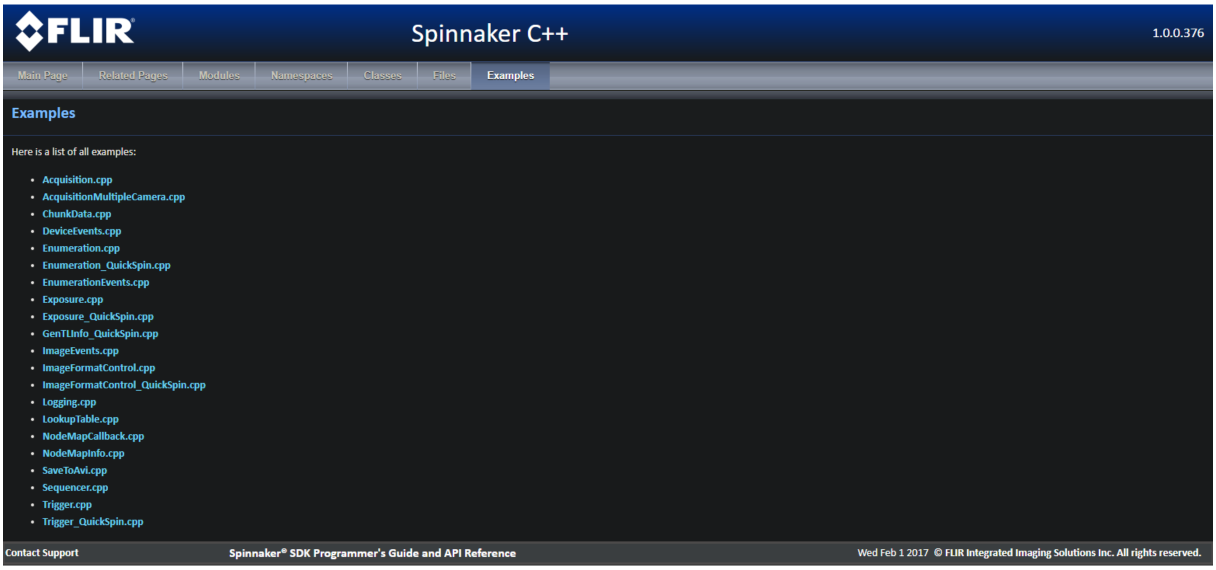
Cameras	Serial	Model	Custom Name
□			
☑	14426535	Flea3 FL3-U3-13...	

14426535 - Flea3 FL3-U3-13Y3M

Property	Auto	Write value	Read value	Slider
Brightness	☑	0.000	0.000	
Gain	☑	0.000	0.000	
Exposure	☑	1.414	0.837	
Gamma	☑	1	1	
Shutter	☑	1.002	0.045	
Pan	☑	0	0	
Frame Rate	☑	150	150.0	
Auto Trigger	☑	0		

Figure 4 the original Camera Manager project

The next step was now to discover the Spinnaker API which is the new one. So we just installed the SDK on our PC. Then we find the API documentation and tried to familiarise with it. We also found code examples but we were not able to use them in Qt creator, but if I just compiled and launch them in the Ubuntu terminal they were working. So the Spinnaked API was installed and work correctly and the issues has to be from Qt Creator. I have tried few things to make it works but with no succeed. During this time I was working on Ubuntu, one of the other students were working on setting up the project on Windows. He found that we needed the 2015 version of visual studio or it will not work. So he send me what he found and what I needed to install it on my PC. So the project worked well, so I tried to use the spinnaker API and to use a code example in Qt creator for windows.



Examples

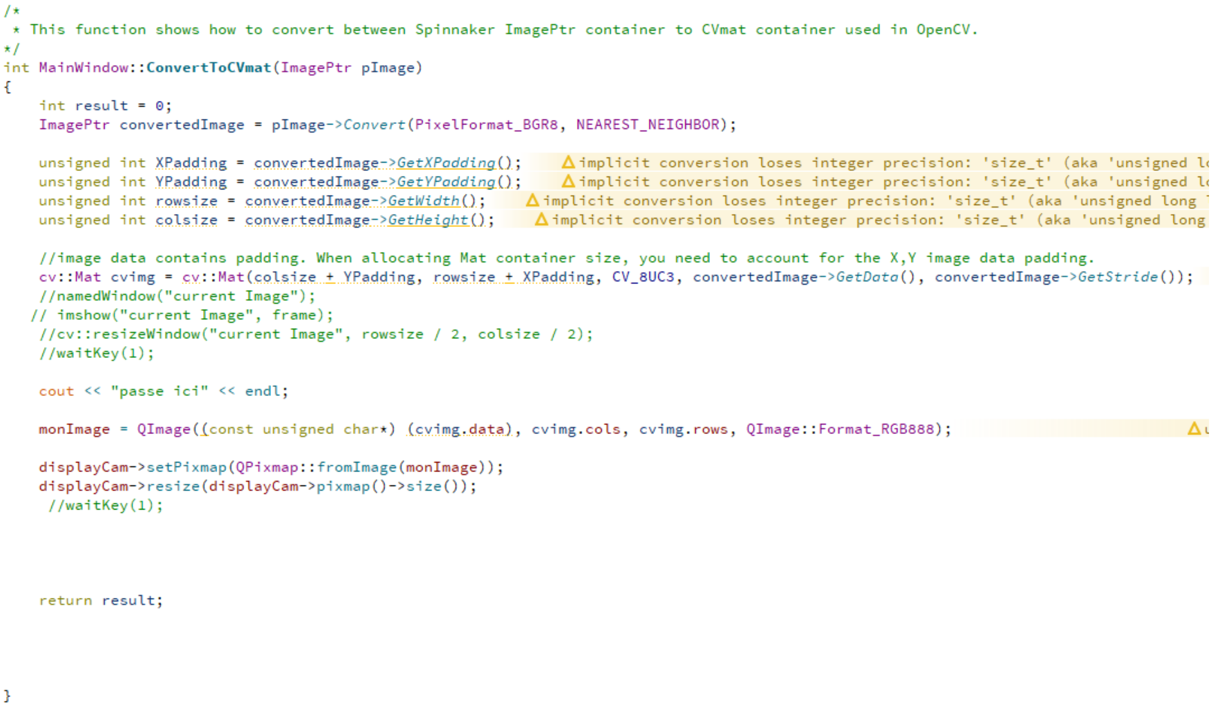
Here is a list of all examples:

- Acquisition.cpp
- AcquisitionMultipleCamera.cpp
- ChunkData.cpp
- DeviceEvents.cpp
- Enumeration.cpp
- Enumeration_QuickSpin.cpp
- EnumerationEvents.cpp
- Exposure.cpp
- Exposure_QuickSpin.cpp
- GenTLInfo_QuickSpin.cpp
- ImageEvents.cpp
- ImageFormatControl.cpp
- ImageFormatControl_QuickSpin.cpp
- Logging.cpp
- LookupTable.cpp
- NodeMapCallback.cpp
- NodeMapInfo.cpp
- SaveToAvi.cpp
- Sequencer.cpp
- Trigger.cpp
- Trigger_QuickSpin.cpp

Contact Support Spinnaker® SDK Programmer's Guide and API Reference Wed Feb 1 2017 © FLIR Integrated Imaging Solutions Inc. All rights reserved.

Figure 5 Spinnaker documentation and examples

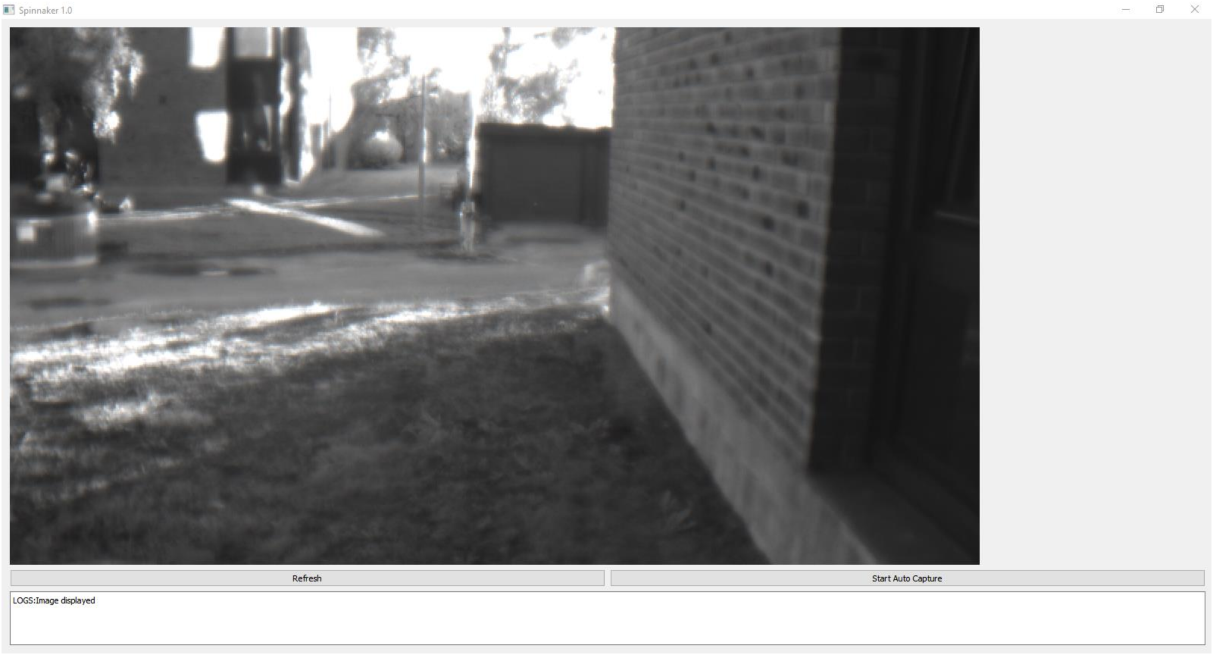
I was surprised that it works perfectly on it. I decided to work on windows for now, to continue to work on the API and we will try to understand why it don't work on Ubuntu later. I have remarked an example which set up the camera, acquire 10 images and save them as files. I was thinking it was good beginning cause it taking images and streaming videos are the main objectives. So I worked on this example to understand it nicely, and then I tried to make my own Qt application which display the captured image directly inside the window. I success creating my own window and adapting the code but it wasn't possible to display the image. In fact, the only way to display an image with Qt is to set a label inside a window and to use a pixmap from a QImage. But there is the problem, the image returned by the camera is an ImagePtr which is specific to Spinnaker. So we had to find how to convert it, or to find a tools which allows us to display the image. After few research on internet, we found something which may work with the OpenCV librairies. We found a function from OpenCV which can convert the image in ImagePtr to an OpenCV matrix which is CVmat.



```
/*
 * This function shows how to convert between Spinnaker ImagePtr container to CVmat container used in OpenCV.
*/
int MainWindow::ConvertToCVmat(ImagePtr pImage)
{
    int result = 0;
    ImagePtr convertedImage = pImage->Convert(PixelFormat_BGR8, NEAREST_NEIGHBOR);

    unsigned int XPadding = convertedImage->GetXPadding();      △implicit conversion loses integer precision: 'size_t' (aka 'unsigned l
    unsigned int YPadding = convertedImage->GetYPadding();      △implicit conversion loses integer precision: 'size_t' (aka 'unsigned l
    unsigned int rowsize = convertedImage->GetWidth();     △implicit conversion loses integer precision: 'size_t' (aka 'unsigned long
    unsigned int colsize = convertedImage->GetHeight();     △implicit conversion loses integer precision: 'size_t' (aka 'unsigned long

    //image data contains padding. When allocating Mat container size, you need to account for the X,Y image data padding.
    cv::Mat cvimg = cv::Mat(colsize + YPadding, rowsize + XPadding, CV_8UC3, convertedImage->GetData(), convertedImage->GetStride());
    //namedWindow("current Image");
   // imshow("current Image", frame);
    //cv::resizeWindow("current Image", rowsize / 2, colsize / 2);
    //waitKey(1);

    cout << "passe ici" << endl;

    monImage = QImage((const unsigned char*) (cvimg.data), cvimg.cols, cvimg.rows, QImage::Format_RGB888);

    displayCam->setPixmap(QPixmap::fromImage(monImage));
    displayCam->resize(displayCam->pixmap()->size());
     //waitKey(1);



    return result;


}
```

Figure 6 OpenCV convert function

This function convert the original image in ImagePtr in the right pixel format and then extract data from this image. After that, the data extracted is used to create a QImage and then we just have to display this QImage in a QLabel using QPixmap. Now, with this function, I have been able to update the project I was trying to build and I succeed to create a program which create a window and whenever a “refresh” button is clicked, the camera acquire an Image and display it inside the window. This was a great step, but the next objective is to display a video stream inside a window. Few time after that, one of the others student success to display a “stream” using a timer which call the acquisition again and again and display the image taken. But the framerate was not good at all and it was quite buggy as well.

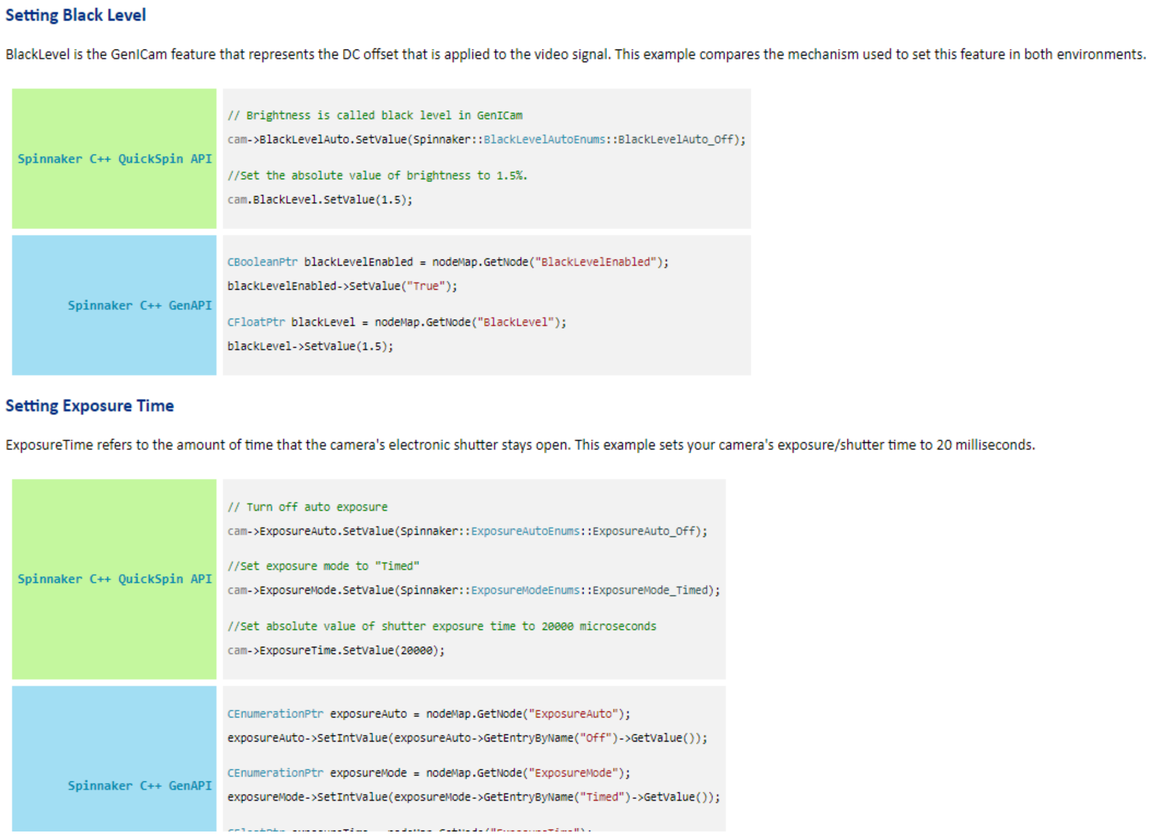


LOGS:Image displayed

Figure 7 First project displaying a video stream

After that, the next step is to be able to have a fluent stream and to display more stream for each camera which are connected at the same time. I checked the original project to have a look on what they were doing about this. I found that they were using the multithreading, which consist to provide multiple threads of execution concurrently, supported by the operating system. In fact, it is a small part of code which can be executed aside of the main program. I have never used threads so I had to learn about what it consist and how to use it. So, like each time, I just searched on the internet and I followed online courses and forum discussions. I made few little projects just to understand threads and then I decided to go further. I decided to use the project of my other colleague which was the first step to display a video stream. My idea was to get his code which working, add some QLabel to be able to display more stream and to launch a thread which acquire and display the image when it detect a camera. Unless that, I get some errors. In fact, it seems that I can't be able to launch the camera acquisition inside a thread. I continued to search about how to solve this problem but I never found a solution. Few times after, we found a way to do it without threads, so we just left aside this part which was complicated while we had a simpler way to do the same.

The next important step, was to be able to use de camera properties. So I started to work on it right after the threads. During the Spinnaker discovery, I had taken a look to the documentation while searching examples. In this documentation there were some littles parts of code which shows how to get and modify camera properties such as Gain, Black Level, Gamma, Exposure, etc.



Setting Black Level

BlackLevel is the GenICam feature that represents the DC offset that is applied to the video signal. This example compares the mechanism used to set this feature in both environments.

Spinnaker C++ QuickSpin API

```
// Brightness is called black level in GenICam
cam->BlackLevelAuto.SetValue(Spinnaker::BlackLevelAutoEnums::BlackLevelAuto_Off);

//Set the absolute value of brightness to 1.5%.
cam.BlackLevel.SetValue(1.5);
```

Spinnaker C++ GenAPI

```
CBooleanPtr blackLevelEnabled = nodeMap.GetNode("BlackLevelEnabled");
blackLevelEnabled->SetValue("True");

CFloatPtr blackLevel = nodeMap.GetNode("BlackLevel");
blackLevel->SetValue(1.5);
```

Setting Exposure Time

ExposureTime refers to the amount of time that the camera's electronic shutter stays open. This example sets your camera's exposure/shutter time to 20 milliseconds.

Spinnaker C++ QuickSpin API

```
// Turn off auto exposure
cam->ExposureAuto.SetValue(Spinnaker::ExposureAutoEnums::ExposureAuto_Off);

//Set exposure mode to "Timed"
cam->ExposureMode.SetValue(Spinnaker::ExposureModeEnums::ExposureMode_Timed);

//Set absolute value of shutter exposure time to 20000 microseconds
cam->ExposureTime.SetValue(20000);
```

Spinnaker C++ GenAPI

```
CEnumerationPtr exposureAuto = nodeMap.GetNode("ExposureAuto");
exposureAuto->SetIntValue(exposureAuto->GetEntryByName("Off")->GetValue());

CEnumerationPtr exposureMode = nodeMap.GetNode("ExposureMode");
exposureMode->SetIntValue(exposureMode->GetEntryByName("Timed")->GetValue());

CFloatPtr exposureTime = nodeMap.GetNode("ExposureTime");
```

Then I decided to start with that and to work on the exposure example to understand how these properties works and how to set a value. I tried with the first one mentioned in the documentation, the Black Level. This one was working in a standalone project I have made so I quickly decided to try to create my own project. The idea of this project was to have a stream in a window, and below, some sliders to be able to modify the property value. Then I started to create the core of the project, with the windows creation and the stream display which I was already mastering. This was not quite difficult and then I started adding the code used for the properties. The way I have done it was working with the Black Level. After that success, I continued to try to get more of the properties working. The next one was the Gain and I don’t get any problems as well. But, it was not the same deal with the other properties. They were all creating the same issue which was saying that the node corresponding to the property was not writable. In fact, the system of property was not working with only something like cam.setGain(value). This only works on windows but not on Ubuntu, so we decided to use the other way, the nodes. There were a lot of not used by GenICam, a generic interface for cameras used by Spinnaker. It allow us to get nodes and to read or write values on it. The nodes I tried to access were existing and were readable, they were only not writable. Again, I have read all the GenICam documentation to find an explanation. I finally find something about some function like the acquisition one, which may set some other node only readable. Then after that, I was minding if I changed the place where I set the property it will work, but with no success. After a meeting, they gave me the idea to try to stop the acquisition just the time to change the property I wanted to modify. I tried few ways to do it but I still get the node permission issue. After that, I sent my work to one of the others student to take a look on that problem. So during this time I adapted the main project to work on Ubuntu. About the properties, the other student find that it was because the cameras we had were not completely compatible with the API but the properties works well with the new cameras. So he used what I have made to implement it into the main project.

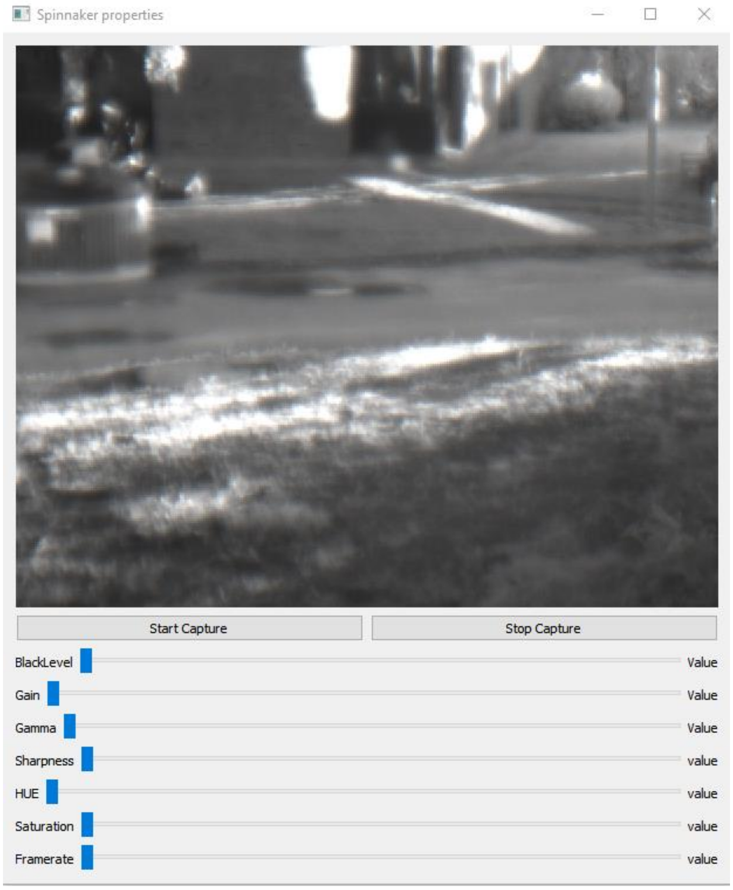
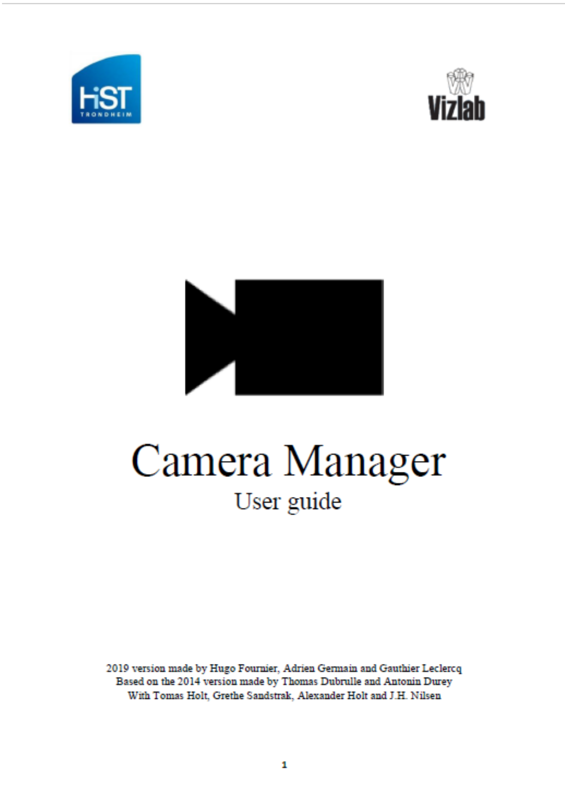


Figure 8 Project to modify image properties

After that, the project was nearly finished and it still only few bugs. The main one was were that some squares appeared in the display window using new cameras. In fact, just renounce using openCV display images because they were using openGL in the project and they had a function "sendframe" which allow us directWe were aware that it must be because the camera were in color while the display was in grayscale. So it must have a conversion somewhere with openGL but we haven't find it. But I was thinking, unless that, if we can be able to send the same type of image to openGL than the old cameras, the problem may be solved. Then right after that, we tried a conversion to mono we seen before and it worked perfectly. Here was the real end of the project, we finished it by making commentaries on code and make some documentation about it with the two guides our tutors needed.

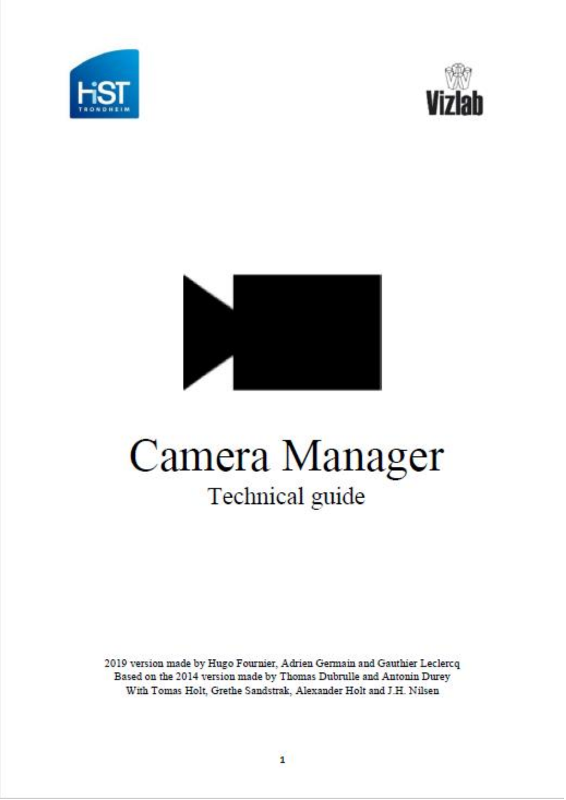


Camera Manager
User guide

2019 version made by Hugo Fournier, Adrien Germain and Gauthier Leclercq
Based on the 2014 version made by Thomas Dubrulle and Antonin Durey
With Tomas Holt, Grethe Sandstrak, Alexander Holt and J.H. Nilsen

1

Figure 9 Camera Manager User guide



Camera Manager
Technical guide

2019 version made by Hugo Fournier, Adrien Germain and Gauthier Leclercq
Based on the 2014 version made by Thomas Dubrulle and Antonin Durey
With Tomas Holt, Grethe Sandstrak, Alexander Holt and J.H. Nilsen

1

Figure 10 Camera Manager Technical guide

Difficulties encountered and what it brings me

From the beginning, this project were not quite easy. It deals with the fact that it is an entirely IT project while my main formation deal with electrical engineering. I have made some project during my formation where I had to code but it was not as advanced as this one. So the fact that I had to learn a new programming language in few days while I am not really a specialist in programming. However, I have made some programming and I learned some notions of java and c during my electrical engineering formation and it has helped me a lot. Then, there is also that in this king of project, there is always something wrong and it don't work well at the first try. So I had to carry on sometime a lot of time searching for a solution, but I think this is a good mental training. It was also important to keep a good communication with the two other student to help us together and always going further in the project.

Unless it was difficult, I think this project brings me a lot. In fact, I have seen another way to work because I had never worked on only IT project. I also learned and worked with a popular programming language which is C++. It may be an advantage to already have seen it because I wanted to go to embedded systems for my engineering studies and it is quite used. I think it also brings me some project management skills because it were the longest project I ever made and I was in total liberty and self-directed in my work. As a consequence it brings me a lot of independence and to be able to make the right choice at each moment of the project. And obviously it helped me improve my English mastering.

Conclusion

In closing, even if I encountered few difficulties, I tried my best to go further and to bypass them. I had to adapt quickly to another type of living, to another type of project that I had never experienced. In fact, thanks to that, I learned a lot about some things. I discovered how people are living here, the Norway landscape which is unique.

For sure, I also improve some of my skills and I have develop new ones. I think I have improved my English mastery, mostly the oral part by talking and living with other nationalities people. I also develop my project organisation skills, communication skills and programming skills. The project, unless to be difficult for me, was quite interesting because I had to go out of comfort zone.

I had to learn by myself and to try my best unless I were not as trained as my two other colleagues for this project. I think that was for me a good opportunity and experience, about never giving up, always trying and work to get to the end and what you wanted to achieve.

Sources

I only used Wikipedia and the NTNU website for the University presentation.

www.wikipedia.com

www.ntnu.no

And I also used the online spinnaker documentation to take screenshots

http://softwareservices.ptgrey.com/Spinnaker/latest/index.html