Assignment: DICOM Discovery

1 Introduction

In this assignment, you will start exploring DICOM files and their usage through DCMTK API. At the end of the assignement, you should obtain a Qt based interface allowing to visualize images contained in a DICOM file and allowing to select a specific window.

This assignment requires that you actively explore various documentation listed on the teaching unit homepage. In order for you not to loose time, a simple skeleton of Qt application with the required libraries is provided.

The skeleton can be built at CREMI using the following procedure:

```
mkdir build && cd build && qmake --qt=qt5 .. && make
```

2 Accessing information in DICOM files

The initial skeleton has a simple menu with two possible actions:

- openDicom: It does nothing currently, you have to implement it to open a Dialog window allowing the user to select a DICOM file.
- showStats: It should display information about the data currently loaded. Initially it should open an information message box with a lot of TODO.

Start by implementing the openDicom method in order to store the data contained in a DcmFileFormat object. Then, try to find the Tag associated with the TODO in the showStats message and complete the message shown. At some point, this will require to access the image data from your file, depending on the transfer syntax used for the dataset, this might require to load codecs before.

If you want, you can add other relevant information to the messages displayed.

3 Displaying images

3.1 Basic display

The next step is to allow visualization of the data contained in the DICOM file in your Qt window. This will require to:

- Extract an image representation from the DICOM file.
- Add a container to display an image and set its content appropriately.

While doing this, keep in mind some basic principles:

- When the window is rescaled, the image should be updated as well.
- The original aspect ratio of the image should be conserved.

3.2 Adjusting window

A key element when visualizing data is to be able to focus on a range of density relevant to the application. Using the `setWindow` method will allow you to increase the contrast on the elements you desire to highlight.

Figures 1 and 2 represents the same data but with different windows.

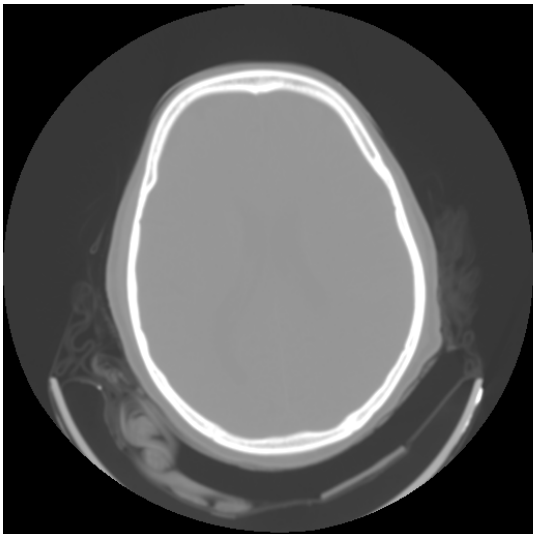

Figure 1: Visualization with a large window

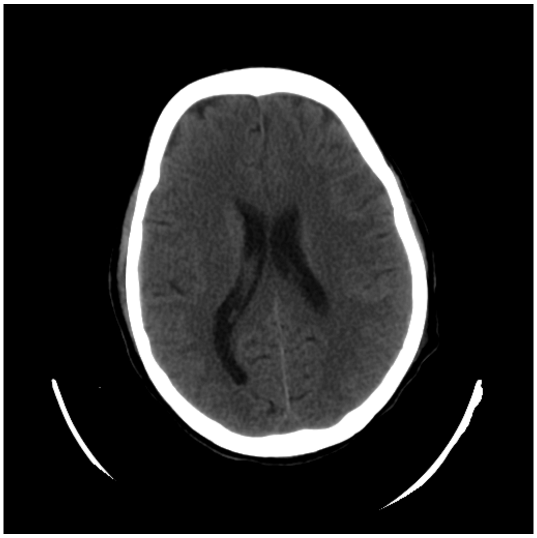

Figure 2: Visualization with a narrow window

3.3 Save images

Finally, implement a method allowing you to save the image currently displayed on screen in a classic image format. Make sure to bind it to the usual shortcut for saving files.