Task 1 – Multi-Channel Signal Viewer

Introduction: Monitoring the vital signals is a crucial aim in any ICU room.

Description: Develop a multi-channel signal viewer that has the following features:

- The user can browse his PC to open any signal file. Each group will need to provide samples from three different medical signals (e.g. ECG, EMG, EEG,…etc). Each signal type should have an example for normal signal and abnormal signal.
- Your application should contain one main graph. When the user opens a signal file, it should show up on your graph in the cine mode (i.e. a running signal through time, similar to the one you see in the ICU monitors). Do NOT open a signal in a static mode.
- The use can manipulate the running signals through UI elements that provide the below function:
    - Change color
    - Add a label/title for each signal
    - Show/hide
    - Control/customize the cine speed
    - Zoom in/out
    - Pause/play
    - Scroll/Pan the signal is any direction (left, top, right, bottom). Scroll is through sliders, and pan is through the mouse movements.

  During these manipulations, you need to take care of the boundary conditions! Intuitively, no scroll/pan should be allowed before your signal starts or after it ends or above its maximum values or below its minimum values. No user expects to see an empty graph coz he scrolled the signal too much to the top for example. Note: Ofcourse, all manipulations will be applied on all the opened signals (viewed or hidden).
- For each opened signal, the user can visualize the signal spectrogram (search for the term and how to generate it) image somewhere beside the main graph (not as a popup window!). Please, show your design skills to produce a nice-looking combination between the graph, the spectrogram, and the UI elements.
    - The user can control/customize the relative size of the signal graph and the spectrogram image via dragging a splitter or any other convenient method (Do NOT pop up a window that asks the user about the relative size!!!).
    - The UI can display several signals but only one spectrogram (for one of the displayed signals). The user should control which spectrogram to display via a combobox or menuitem that shows the labels/titles of the signals. When the user selects the label/title of the signal s/he wants, the spectrogram should be updated accordingly.
    - The color palette that is used in the spectrogram through a combo-box/drop-menu that has 5 different palettes to use from. You can use some standard palettes from your library. You do not need to create your own palettes.
    - The user can control the level of details shown in spectrogram by changing the range (i.e. min and max values) of the pixels intensity that are displayed. The use can control this through two sliders: one for the min value and one for the max. Each slider should go from min and max values of the pixel intensities of the spectrogram. Please, check the function imcontrast in Matlab to understand this feature before implementing it in your program.
- Exporting & Reporting: For the sake of reporting, the user can export the current status of the graph, spectrogram along with some data statistics on the displayed signals to a pdf file. You need to generate the pdf contents via the code. i.e. Do NOT take a snapshot image and convert it to a pdf file!
    - Data statistics can be mean, std, duration, min and max values for each signal. These numbers should show up in a nice table in the pdf file. The table can have the signals in different rows and the values in different columns.

Code practice:

- Use a proper variable and function names. If I do not understand what your variable is roughly doing without asking for your explanation, then this is NOT a proper name! Examples for non-proper names: x, y, counter, ss, ii, s_i,…etc. Each non-proper variable or function name will be penalized with -20% of the whole task grade.

General Notes:

- This is a task for an engineer who had had a reasonable experience with software programs. And thus, s/he is expected to provide a convenient, user-friendly software. Do NOT invent a feature or a user-interaction that you had never seen before in another software. Do NOT INVENT but rather IMMITAE what you had experienced before with software. If you feel you are very smart that your feature is completely new and no one thought about or saw it before, then we do NOT want to see it either! During delivery, you will always be asked this question "where did you see this feature before?" either for features related to signal viewers or related to dealing with the computer or software in general.
- If you are new to signal viewers, try to download a couple and experience them. There are tons of free downloadable viewers on the internet. Not seeing a viewer before is NOT an excuse for the previous note.