DSP Final assessment research project

Rules:

- Each group consists of 3-4 members. All group members will have the same evaluation.
- Each group should work independently. While you can always consult and brainstorm with anyone outside the group, each group MUST write their own code! Code similarity among groups will NOT be tolerated!
- The project is split into separate problems, each problem deals with specific skill in the DSP realm. The overall grade is 100 marks which will be scaled to the full mark of your assessment grade. Note that there are more than 100 marks available in the assigned problems. So, choose the problems you feel more comfortable with.
- For each problem, you are required to submit your full code along with representative screenshots for your results and with any necessary short comments on the results. Any mismatch between the submitted code and representative results will be considered as NO submission for this problem. The GitHub link for submission will be sent separately to the class.

[20 marks] Problem 1. In a time of pandemic, data is the most crucial factor to overcome the pandemic. It is important not only to collect the data but rather to visualize it in a way that would help decision makers to grasp the underlying information. COVID-19 data (#cases, # deaths and # of recovered cases through time and in the different countries) has been publicly available these days. Search for the data, download it and develop your own application to visualize it using the following methods:

A. Animated bubble graph: x-axis is the number of deaths, y-axis is the number of recovered cases, the size of each country bubble is the # of its cases.
B. Animated maps graph: Using some colormap, each country in each frame is assigned a color that indicates the # cases or # number of death cases in this country. The user can click on any country on the map to plot its number of cases/deaths through time.
C. Animated Sorted chart (i.e. bar rank): each country is represented as bar on the chart. The length of the bar is the #cases or #deaths or both. The bars are always sorted (according to #cases or #deaths. Option in your UI) while running through time.
D. D. Animated bubble graph tailored to an idea of your own. i.e. you can choose the different axis (x, y, bubble size, bubble color) to elaborate or emphasize some specific observation/idea/conclusion based on this data.

All animated figures can run through time (i.e. days from a reference starting day, let's assume it is 01/01/2020) through a stop/resume button. All figures can be exported to a video file through an Export button. Some resources to check (gapminder project, Hans Rosling Ted's talk).

[20 marks] Problem 2. Jpeg compression is a direct application for Fourier transform as a sparse domain. Develop your own decoder for Jpeg images where the user can browse for an image and display its decoding process progressively, i.e. step by step, till it reaches the final image (check the difference between baseline and progressive jpeg decoding here and here along with a good discussion about them here). Your program should read the image, decode it progressively as if the image data is fed to your program chunk by chunk, then display the output of 8-10 of these steps. i.e. your UI should have 8-10 complete images starting from the one encoded by a very low amount of data to the one encoded with the full data. Note this is a decompression/decoding task NOT a compression/encoding one!

[20 marks] Problem 3. It turns out that most of the musical instruments can be modelled and their sounds can be synthetically generated on your computer using a mix of some harmonic waves along with some filters according to some modeling equation (regular or differential). Select two musical instruments, research them, and develop your own program to simulate them (i.e. your program should generate a close sound to what this physical instrument generates in reality). Your UI does not have to reflect how the instrument look visually, but it should allow the user to play his/her music through your UI and also give the user some of the instrument's basic options.

[20 marks] Problem 4. Blind source separation aims to extract original unknown source signals from some mixed ones. This is done by calculating an approximate mixing function using only the available observed mixed signals. "Blind" here means that the mixing function of signals which are recorded by microphones is unknown. Independent Component Analysis (ICA) is among the famous tools used for this job. In one of your tasks, you used some software program to separate the music and vocals contents of a song. You are now required to develop your own separation tool. Do you readings and develop your tool using available libraries for ICA (and/or any other tool you find useful). Your tool should be able to separate the following signals into their components:

A. Songs into music and vocals.
B. Two signals of your own choice. You can choose from cocktail party voices mixture, abnormal ECG signal (i.e. with arrhythmia), Event-related fMRI, or any other signal you feel interested in. You research the topic, find an example for each signal, download it and feed it to your program. Generate your screenshots and put a description for the signal you obtained and/or the problem you solved along with your comments on the results you are attaching.

[40 marks] Problem 5. Like voice signals, cough is a mechanical wave generated by the respiratory system and get spread around the individual human. Unfortunately, this wave carries tiny saliva droplets while travelling and thus it becomes important to address in pandemic times since these droplets are the easiest vehicles to move viruses and microbes from person to another. Several efforts have been done to simulate and visualize how cough travels when generated. Check these examples (Example1, Example2, Example3) to educate yourself about it and search for the simulation equations that govern the cough travelling/propagation. Then develop a program to simulate the cough propagation in air. Your program should have the following features:

A. [3 marks] A 2D viewport that visualizes the propagation in the coronal plane through time. There should be a slider that determines how far the simulated plane is located from the human source.
B. [3 marks] A 2D viewport that visualizes the propagation in the sagittal plane through time. There should be a slider that determines how far the simulated plane is located from the center point of the human source.
C. [3 marks] The user can click on any point in the above two figures to see the wave/signal evolution at this point in a separate 2D graph (wave strength vs time).
D. [3 marks] Face mask acts as a filter that affects the speed and strength of the cough wave. Simulate the mask effect on all the above viewers. i.e. your UI should have a checkbox for with or without a face mask.

You program should have some UI elements to adjust the main parameters in the simulation equation. In A and B, a timer indication should be visible on the viewer while the simulation is running, and the viewer can export the simulation as a movie file.