Final Projects

DUE: Friday, 2016-12-16 at 10:00 PM (when our final exam block time would end)

- You will be working in your assigned group. [1] Please create a repository called `project_netid_netid` (fill in netid of each student member). Be sure to add Dr. Palmeri (`mlp6`) and Brenton (`bnk5`) as a Master access-level member.
- Use all good git repository management practices that have been promoted all semester.
- Create Issues (that are associated with Milestones) for all development tasks on the project, and **assign a specific group member** to each task. While this is a group project, each group member will be graded individually based on their contributions to the project, so strive to have even effort distribution, as represented by these issues. **Be sure to associate commits with specific issues.**
- Use all good python coding practices that have been promoted all semester, including PEP8 style compliance.
- **At least one "core" computational aspect of your codebase needs to be implemented in C/C++ and interfaced as an importable module into Python using SWIG.**
- Create an annotated tag (`v1.0.0`) of your final version.
- Choose one of the following project topics:

  1. Develop software that identifies the P, R and T events on a per-beat basis in an ECG signal. This software will:
     - Read in time and voltage data from a Matlab (v5) file over a finite period of time,
     - Display/save a plot of the ECG signal with each P, R, and T event indicated,
     - Save an output file that stores all of the absolute times for each P, R and T event.
  2. Develop software that augments your ultrasound B-mode image generation assignment to:
     - Have an interactive GUI using either Tkinter or Qt,
     - Choose the JSON and binary datafile to load in the GUI,
     - Provide the ability for the user to interactively change the logarithmic compression level, TGC, and other layers of image optimization (e.g., histogram equalization) and re-render the image.
     - Provide a 'Save' option to save a PNG, JPG or TIFF file based on auto-discovery of the file extension.
  3. Automated Cervical Cancer Screening Project (please see attached PDF). This project will involve implementing a Support Vector Machine (SVM), which is greatly facilitated using the `scikit-learn` package.
  4. In addition to generating B-mode images, ultrasound can be used to generate M-mode images, where data are acquired at the same location serially through time. These M-mode data are used to estimate motion at that spatial location through time. A common application of this is to track the motion of ventricular and septal walls of the heart throughout the cardiac cycle. You will develop software that:
     - Reads in metadata in JSON format and RF data from a binary file,
     - Estimates the localized motion of cardiac ventricular and septal walls in the M-mode data using RF cross correlation and phase-shift estimates in demodulated IQ data,

[1] `https://gitlab.oit.duke.edu/medical-device-software-design/rw_data`

    - Plots and saves the localized motion in a developer-decided format.
5. Group-defined project. You are welcome to propose your own project. Please submit a project proposal to me by email that includes:
    - Overview of software (clinical application)
    - Functional specifications
    - Data to be input
    - Expected algorithmic implementations
    - Data to be output

- Test data will be posted for the ECG, B-mode, M-mode and cervical cancer screening projects.
- Grading criteria: *You should approach this final project as an opportunity to show a potential future employer an example of your software development skills.*
    - Git Repository
        * Issues/Milestones [10%]
        * Commits are discrete, logical changesets [10%]
        * Branching & Merging [5%]
    - Modular coding [10%]
    - Avoidance of hard-coded variables; robust functional input for algorithmic control. [10%]
    - Full unit test coverage of all functions, with passing CI build[2] [20%]
    - Logging: INFO, DEBUG, ERROR [10%]
    - Sphinx documentation for each module/function [10%]
    - Handle and raise exceptions [5%]
    - Functionality [10%]

[2]Dr. Palmeri will enable gitlab runner for your repository.