CS 251: Lab 11 Python GUI Programming

- Handed out: 03/10 Due: 27/10. 10 PM.
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Objective

Our final code warrior task is to do GUI programming with python. There are several ways of doing this – with OpenGL, wxPython, pyQT and friends, and Tkinter. Each of this method has advantages but at this stage we are going to use either Tkinter or the Django framework.

Related Tasks

Install/Update Python (version 3.4) on your machine. Download the closing rank cut offs, and also programme code from the usual directory.

Learn how to use Tkinter.

Background

Your goal is to create a python GUI interface that allows a high school senior graduate (XIIth pass) engineering college aspirant to fill her college preferences. There are about 100+ institutes Centrally Funded Technical Institutes (CFTIs) who offer about 500+ programmes. As you know from Lab 10 the criterion for selection is either the rank list from JEE(Mains)+College combination, or the JEE advanced rank list.

Task Set 1

The data directory contains the actual data of JEE-2012 opening and closing rank for the IITs for different branches and categories. The file contains statistics in xls format that you will convert to csv using, say, LibreOffice (we don't care). The csv uses unique programme codes.

The data directory also contains a pdf file that describes these programme codes. (It also contains seat capacities in various IITs).

1. Create a program `update.py` that reads the pdf file and creates a csv file that contains
   - College name (e.g. IIT Bombay)
   - Branch name (e.g. Electrical Engineering)
   - Course code

   Save the new table as `data_u-2012.csv`.

2. Write a python program `main.py` to create a `Tkinter` window. Set a relevant size and title for the window. Enable the user to do the following in `main.py` using the closing rank data file.
   - Create an interface that enables her to enter a rank to see prior year opening and closing rank for a particular `(college, branch, category)`. (Recall from Lab 10 that the category could be any of birth category or any other special category). Use the output of the `update.py`.
     You can do this task in any way you want. For example, use colors (say red if a combination is not available, green otherwise) based on the data given.

- Create a form where she can fill her preferences of `(colleges, branch)`. You should also collect enough information so that the comma separated preference list of Lab 10 `choices.csv` can be used by any valid Lab 10 Java program.

  Create a button `Submit` which when clicked saves these values to a file `mydata.csv`. **Note that the candidate can fill any number of preferences.**
- Enable the candidate to enter her ID and retrieve relevant information from the file you have created from a previous save, and display them.

Files To Submit

To be completed

How We Will Grade You

The tasks you need to do have been described. However we will grade you depends on whether you do the pro or the free version.

For the free version

- For the first task, you may not be able to fully parse the pdf file using a completely Python solution. See http://css.dzone.com/articles/pdf-reading. Never mind, do the best you can, for example, by using pdftotext.

  We want to see some effort in this direction, but in the worst case, simply input the text in a datafile.csv and proceed.
- We expect Task 2 to be done as per specification.

For the pro version, you don't have to do the Tkinter stuff. Instead,

- For the first task, we are expecting that you can read some pdf file if not "this" particular pdf file. (All pdf file creation is not equal. For example, you might find the output of pdflatex easier to parse.

  We want to see some effort in this direction, but in the worst case, simply input the text in a datafile.csv and proceed.
- For the second task, you will have to show your creative aspects and create a Web application (using, say, the Django framework). Here are some ideas in sorted order of wishlist. Note that we do not expect all of these – these are just some ideas but we have divided them into classes.

  - Visualize the closing ranks in a more intuitive way. One should be able to visually see what is available based on ranks entered with a collapse by institute, or by department, or by geographical location. (Consider D3.js).
  - Allow bulk writes (e.g., user may say, "mark out all CSE departments first in all IITs then all EE departments in all IITs, then all ME departments in older IITs") and so on.
  - Implement NITs also. We will provide data similar to what has been provided for the IITs (we are looking out for them). For example, if a rank list of IITs and NITs is given (e.g. India Today ratings) then bulk writes can be done.
  - Any other cool idea

  The second set of ideas are

  - Helps candidate decide using "what if" analysis. For example, if CSE is suddenly less popular this year and Mechanical is more, then how do things change

  - The system reacts to dynamic data. For example, if about 1000 people are on the site, your system will show that so many people are online with approximate (average) rank as "x" with a standard deviation of y. If the number of people is low, then it will not use meaningless averages.
  - Implement robust security measures so that no on can crack the site and change the data
  - Stress test the site.
  - Enforce a login process. Implement usual stuff like retrieve lost password and so on.
- Create a video that show cases your project.