Team Project Specification

Due: 11:59 P.M. Wednesday, December 3, 2014

"**AggieSnap**"

The project is to design and write a C++11/FLTK picture database program with a graphical user interface. The program should start with a description of what it does and then allow the user to:

1. Get a picture from a filename. (The file might have come from a cellphone camera.) You only need to accept file extensions of jpg, gif, jpeg, JPG, GIF, and JPEG (unless you want extra credit—see below). Allow the user to recover from entering an incorrect type of file, a file that can't be opened, etc.
2. Get a picture from a URL, using wget as follows:

system((string("wget ") + URLstring + " –O " + filename\_string).c\_str());

to download from the URLstring website to a file whose name is in filename\_string. (See the man pages for more information.) You only need to accept file extensions of jpg, gif, jpeg, JPG, GIF, and JPEG (unless you want extra credit—see below). Allow the user to recover from entering an incorrect file extension, a URL that can't be downloaded, etc.

1. Add up to 5 tags per picture (Family, Friends, Aggieland, Pets, Vacation, or any combination of these).
2. Save the picture to disk, and save information to a picture index file. The picture index file should consist of human-readable text; for example, lines like

(reveilleVIII.jpg,Aggieland,Pets,,,)

since there are only two tags on that picture. This file should be read each time the program is run, so that adding pictures is cumulative. Hint: See the textbook's code for printing Date objects.

1. Browse all pictures. When a picture is displayed, so are its tags. Provide buttons to see the Next or the Previous picture.
2. Find pictures with any combination of tags. Provide buttons to see the Next

or the Previous picture with those tags.

Be creative in deciding how to meet these specifications in an attractive and user-friendly way, but get the basic functionality working before you try to make it too fancy, or you may run out of time!

All user input and output must be through the GUI, not the console window. However, you may use the console window for printing debugging messages for the developers (your team).

EXTRA ITEMS (Teams of 4 must do 2 of these—see below)

Allow the user to change the text of the tag labels and save the new information to the picture index file in place of the old.

Allow the user to delete a picture from the picture index file. (Be sure to rewrite the picture index file.)

Accept two additional picture formats FLTK supports (bm, bmp, pbm, pgm, png, ppm, xbm, xpm, and their upper-case versions).

Extract Exif data (“geotag” GPS location, time and date picture was taken, etc.) from a picture (if present) and display it whenever the picture is displayed.

Use the FLTK function

virtual Fl\_Image \*copy(int W, int H)

to resize an Fl\_Image so that no matter what the picture's original size, it is always displayed in its entirety without being cropped or masked.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is a team project, with approximately three students on a team. The instructor will assign the teams. (Note: If there are any problems with your team assignment,

please talk to your TA or Dr. Daugherity promptly.)

Divide up the code, with each student on a team of 3 doing two of the six items on page 1. If you have a team of 4, the fourth person must do two of the "Extra Items" on page 2 for the team to get a maximum grade of 100.

A team of 3 will receive 5 points extra credit for doing one "Extra Item" or 10 points for doing two. A team of 4 will receive 5 points extra credit for doing a third "Extra Item" or 10 points for doing a third and a fourth.

You must use at least two C++11 features, such as auto and range-based for. You must write at least two classes of your own, with separate header and implementation files. Follow good style, and limit each function to no more than 24 lines (one terminal window). Each team member is expected to have a rough idea of how all the code works, and should be able to explain in detail how their own part of the code works.

Your program should compile and run without change on the Visual C++ environment in the lab or on build.cse.tamu.edu with X windows.

Your program must be submitted both to CSNET and also on a CD or DVD. The project report (described below) should be submitted on paper to your TA, along with your CD. You only need to submit one report and CD and CSNET file per team. Write a report according to the outline below. All team members will receive the same project grade, unless some team member does not do his/her part (see report outline below).

**Important! You must demonstrate your project to your TA or it will not be graded!**

Note about teamwork: Immediately exchange contact information with your teammates and schedule times to meet and work on the project. At a minimum, you have the two hours per week during lab. As meeting scheduling can be difficult, use this lab time wisely! Attendance will be taken during lab, so that complaints of "We could never find a time to meet" will not be taken seriously.

REPORT OUTLINE

The project report must be printed on a laser printer. The report should include the following sections:

1. Team information (team name, members' names, who did what, did each member

do a fair share of the work)

2. Statement of the problem, significance, etc.

3. Restrictions and limitations

4. Explanation of your approach (analysis to choose a strategy for programming the project, how you coded it, etc.)

5. Sample run (screen shots)

6. Results and analysis

7. Conclusions - What did you show? What did you learn?

8. Future research (how your program could be improved or extended)

9. Instructions on how to run your program

10. Listing of the COMMENTED program

11. Bibliography - references used, if any

Credit: A number of the ideas for this project came from Dr. Welch.