cis20.1-fall2007-sklar, assignment I, part 2

instructions

- This is the second part of the assignment for unit I.
- The entire assignment is worth 14 points.
- The first part is worth 4 points and was distributed and worked on in class on Thursday August 30.
- This part is worth the remaining 10 points and will be distributed and worked on in class on Thursday September 20 and Thursday September 27.
- **Both parts together are due on Monday October 1** and must be submitted by email (as below). Please WAIT and send ALL PARTS TOGETHER, not one at a time as they are completed.
- **Follow these emailing instructions:**
  1. Create a mail message addressed to *sklar@sci.brooklyn.cuny.edu* with the subject line **cis20.1 hw1**.
  2. Attach PDF and source code (Java or NetLogo or Scratch) files. Please zip them together using gzip or zip or stuffit or WinZip — but NOT WinRAR or rar, please!!!
  3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work — grrrr!)

1 designing the interface

(4 points total, distribution described below)
Based on what you have learned so far about user interfaces, **design** your own interface for sharing photos that will be able to handle (at least) 20 photos and will provide a creative way for doing 3 tasks with the photos.

Here is a list of possible tasks, or feel free to come up with task(s) of your own:

- annotate a photograph (i.e., associate a text tag with the image)
- search for one photo in one of the following ways:
  - perform a text-based search of annotations (match text tags)
  - perform a date-based search (match dates)
  - perform an image-based search (match images/image features)
- sort the photos in one of the following ways:
  - arrange photos based on similar text annotations
  - arrange photos based on similar dates
  - arrange photos based on similar image features
- manipulate images in one of the following ways:
  - zoom in and out
  - resize
  - change color

This step is the DESIGN PHASE ONLY!!! You should NOT write any code for this step.

- Design on paper (or using a digital drawing tool) what the user interface (i.e., the window(s)) will look like. Include drawings for each screen needed to facilitate each of the 3 tasks you've chosen to implement (as above). *(1 point)*
- Your design should include a text description of how the interface will behave (e.g., "clicking on the **sort** button will rearrange the images in chronological order"). Include descriptions for how each of the 3 tasks will be performed from the user's point of view — what input do you need from the user and what output will the user see/receive? *(1 point)*
- If your interface has multiple windows or screens (it probably will), then you should include in your design a drawing of a tree that shows the logical progression from one screen to another, including an indication of how the user will go from one to another (i.e., by clicking on button X...). *(1 point)*
- Include an error analysis—what kinds of things might the user do wrong? how will your interface respond to user errors? *(1 point)*

2 implementing the interface

(6 points total, distribution described below)
Since **unit I** for this class is focused on **interfaces**, you will only have to implement the user interface portion of the system (you will implement the underlying functionalities later in the next unit).

You can choose to implement your interface in either Java, NetLogo or Scratch. The idea here is to **be creative**. Play with the various interface components available in Java (see my sample programs posted on the class web page), in NetLogo (see the community models posted on the NetLogo web page) and/or in Scratch (see the shared projects on the Scratch web page).

What I want to see here is the user interface only — not the underlying tasks. You don't have to implement zooming, sorting, annotating, etc. (you'll do that for the assignment in unit II). You only have to implement the widgets (e.g., buttons, menus...) that will facilitate the user requesting that the 3 tasks be performed, i.e., moving from a main screen to each of the task screens and back to the main screen. So for example, you could decide to implement zoom, sort by date and annotate. The main screen of your interface could simply have three buttons on it (boring!), labeled "zoom", "sort" and "annotate". If the user clicks on "zoom", then all you have to do for this part of the assignment is to generate a new screen that says something like "the user will select an image to zoom here" and then a third screen that says something like "the zoomed image will be displayed here".

Your implementation should include:

- a main screen *(2 points)*; note that this screen should provide a method for going to each of the task screens (see below), a method for accessing the help screen (see below), and an "album" view of all 20 images in your sample database of photographs
- a screen for each task *(3 points, one per task)*; note that this could be a variation of the main screen and/or it could involve a number of sub-screens (windows?) required to request information from the user (e.g., "select image to zoom")
- a help screen *(1 point)*; this should be easy to find and easy to follow, and should include a text description of how the user can (will be able to) execute each of the 3 tasks