Basic rules for assignment A4

Time spent. We will ask you tell us the number of hours you spent on this assignment when you submit your files. We will announce the mean, median, and maximum time spent. Please keep track of the time you spend.

Work in groups of 1 or 2. A4 may be done with one other person. The A3 handout contains instructions for grouping and also talks about how group members should work together. Those instructions apply to A4 as well.

Academic integrity. Never look at or be in possession of the assignment A4 code of another group, in this semester or a previous one. Do not show or give your code to another student. Do not post assignment A4 code on the Piazza. If our system finds that two solutions are very similar, expect that we will ask you and the other person to come meet with us and explain what happened

Naturally, you may discuss A4 with others, but the discussion should not extend to writing actual code, picking variable names, agreeing on specifications or comments, etc. We check for unexpected similarities and consider it to be a violation of the academic integrity rules if code was shared.

Building on A3. Assignment A4 builds on A3. We will make our solution to A3 available on 10 April, and you may use that as your starting point. If you had errors in your A3, using our A3 may be the best choice.

Keeping Sane. A lot of things are going on in this semester-long project. Here are ways to stay sane: (1) Rather than wait until the last day or two to work on A4, start early and do it in stages. Below, we suggest a timeline for A4. (2) Specify methods carefully, including preconditions; put meanings, with constraints, on fields. (3) Create a JUnit test class, and put tests in for every method you write as you write it.

What A4 is about

One of the expected outcomes of CS 21110 is, that you will be able to design GUIs, but we have had only two lectures on that topic and no homework. This assignment will give you some experience. You have already created an array of species and a distance matrix for them. This assignment is to provide a GUI for them. Your GUI will allow you to click on an image of a species and (1) see the species that is closest to it and (2) recognize, by their background colors, which other species are closest and farthest from the one you clicked on.

Thus, this assignment is a change of pace. We give you directions, as usual, but you may have to do some studying regarding placement of components in a GUI and listening to events on the GUI.

Look at the picture on the next page. It shows an array of images for all the species in directory SpeciesData —the images are the .png files in the directory. In the GUI, someone clicked on Darwins Tortle, so it appears with a red background and its closest relative is Fuzzy Trible. Yellow means closely related to Darwins Tortly, and the greener the background, the more distantly related the species. *Note that the distance data is deliberately wrong. Fuzzy Trible is not the closest species to Darwins Tortle, and the colors in the array are meaningless.*

What to do for this assignment

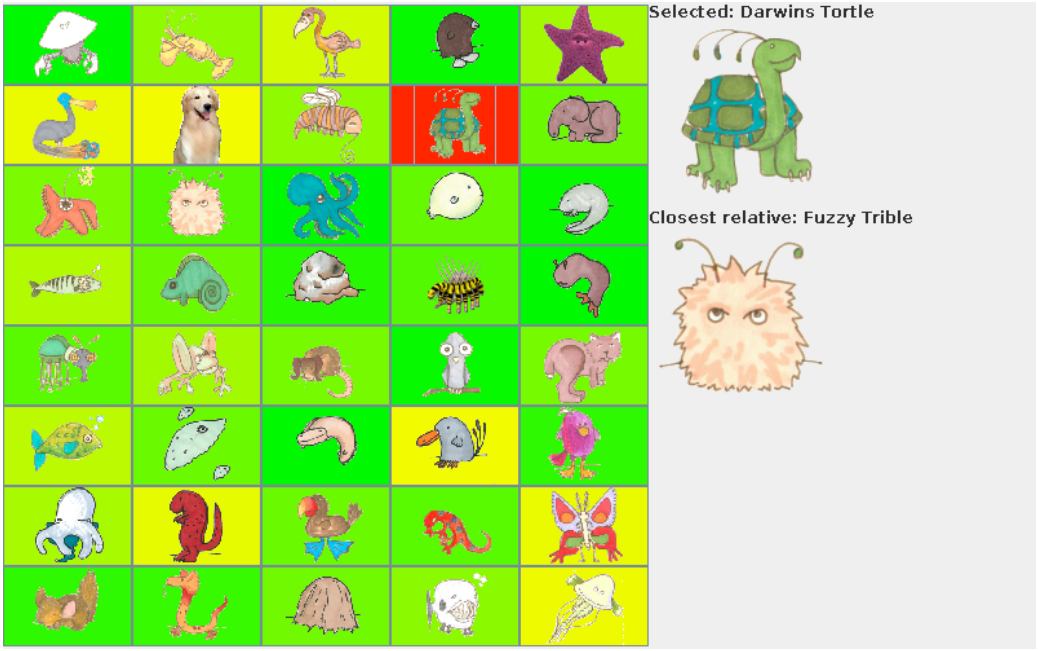
We now describe how to go about implementing this. We give suggested times to complete each part.

Step 0: Set up project. Suggestion: Complete this step by Wednesday, 10 April. Start a new Eclipse project, perhaps calling it "a4". Put in it all your files from A3 (or our solution). Method Main.main, should be as you submitted it in A3 —it should compute the array of Species objects and the species distance matrix and print them.

Download a3.zip from the Piazza or from the CMS. It contains a package a4GUI of classes that implement the GUI. Add this package to your Eclipse build path. Here are two ways to do it

- Import it into the project: right-click (control-click on Macs) the src directory, and choose Import. Import an Archive file (under General), then find a3.jar and import it.
- Open the zip file and copy all of its contents to the project but in directory src.

Place a copy of directory SpeciesData in the project, as you did for A0, A1, A2, and A3.



Step 1. Write class A4GUI. Suggestion: Complete this step by Thursday, 11 April. Create a class A4GUI, which extends class a4GUI.ComparisonGUI. Yes, you can write the package name followed by a period before the class name. Put in a constructor with this heading:

```
/** An instance for n species. n > 0. */
public A4GUI(int n)
```

Look at the specification of the constructor in ComparisonGUI to see how to call it, so that you can write your A4GUI constructor properly. Don't be concerned at the moment with the part of the specification of the constructor that says what to do after constructing an instance of ComparisonGUI.

Now turn to method Main.main and, at the end of it, place this statement:

```
A4GUI a4gui= new A4GUI(30);
```

Run Main (i.e. call Main.main). A GUI will appear with an array of buttons that say simply "image". Change the number 30 to 8 or any other number and run again.

Study class ComparisonGUI. Look at its fields. Their access modifier is protected. This means that they can be referenced in this class, in a subclass, and classes in the same package.

Look at its constructor and note that it basically just calls setup, setResizable, pack, and setVisible. Study setup, which takes care of the comparisonBox on the east (does not work yet), puts in spaces in the center, and fixes the array of images. Study method fixBoxArray. Understand how it (1) uses objects of class Box to create the array of images, how it fills each element of array cell —including in it an object of class ScalingImageButton and its index i. Notice that it registers each ScalingImageButton as an action listener and as a component listener.

Step 2. Fix the comparisonBox, which goes in the east of the JFrame. **Suggestion**: Complete this step by Saturday, 13 April. When the GUI first becomes visible, you are supposed to see to the right (on the east) the labels Selected and Closest relative, but no images. Click any image in the array of images on the left of the GUI and that image and its closest relative are supposed to appear in the east. None of this happens yet.

Your job is to override procedure ComparisonGUI.fixComparisonBox, whose purpose is to add the four necessary components to comparisonBox. That is, write the method in class A4GUI. See the specification of the procedure. Once you fix this, the two labels will appear but no images, since no images appear in the GUI yet.

Step 3. Installing the images in the GUI. Suggestion: Complete this step by Monday, 15 April. Write a second

constructor in class A4GUI that has two parameters: the array of Species (already sorted by name) and the species distance matrix. The constructor should:

1. Call the superclass constructor appropriately.
2. Save the parameters in fields of the class.
3. Populate the GUI with the species images. Do this by making calls on procedure ComparisonGUI.setCellImage. Its specification tells you how to call it.

Now change the new expression at the end of method Main.main to a new expression that calls the new constructor. When you run Main.main again, the GUI should be populated with the images.

Step 4. Responding to mouse clicks. Suggestion: Complete this step by Wednesday, 17 April. Run the program, click on a few images in the table, and note how each click causes a line to be written in the Console, giving the number of the cell clicked. Change this by overriding procedure ComparisonGUI.onSelectCell. That method has a specification, saying what it should do; be sure you place this specification on the overriding method that you write in class A4GUI.

We suggest that you do this in two steps. First, get the images to show up in the east panel. ComparisonGUI has setter methods to help you do this. Once that works, do the second step: Change the background colors of the images to reflect the distance from the selected image.

It is up to you to design and implement a method to color background images to indicate in some way the distance from the selected species. The selected species and its closest relative should stand out in some way. Call procedure setCellColor to change the color of a cell. You may want to look also at class a4GUI.Color. Here are two possibilities, but there are many more:

1. Break the species into 4 groups depending on how far from the selected species they are, with the selected species and its nearest relative in one group, and give each group a different color background.
2. (We did this in our solution, but you may not have the know-how to do it.) Set the colors of all cells using a hue/saturation/value color representation to fade from a bright teal color for the nearest species (which is the selected one, at distance zero), fading through increasingly lighter hues of blue and indigo, eventually fading into white for the farthest species.

Feel free add other features. For example, you could put an explanation in the east comparisonBox of what the background colors mean, or a list of the species and their distances from the selected one, ordered by distance from the selected one. The addition of features will be used for extra credit, as in A2: it will be recorded in a separate column on the CMS and used at the end of the semester in calculating your grade.

Step 5. Prepare submission and submit. Suggestion: Complete this step by Friday, 19 April. The deadline for this assignment is Sunday, 21 April.

1. Prepare a ReadMe file (.txt. .doc, or .pdf) with the following information:
   a. Put the following as the first line, where xx.x is the amount of time you spent on A4:

      Time spent: xx.x hours

   b. If you used our A3 solution, put a comment to that effect on a separate line.

   c. Describe your method of coloring to reflect distance from the selected species. Describe any additional features that you added.

   d. Tell us what you thought about this assignment. Did you learn much about GUIs? Was it helpful to see the species distance stuff usin ga GUI? Etc.
2. Submit on the CMS a .zip file that contains all your source files (*all* the .java files) in your project's default package. Please do NOT include package a4GUI and directory SpeciesData. The name of the .zip file is unimportant.
3. Submit the README file on the CMS.