COSC 1P02 Assignment 5

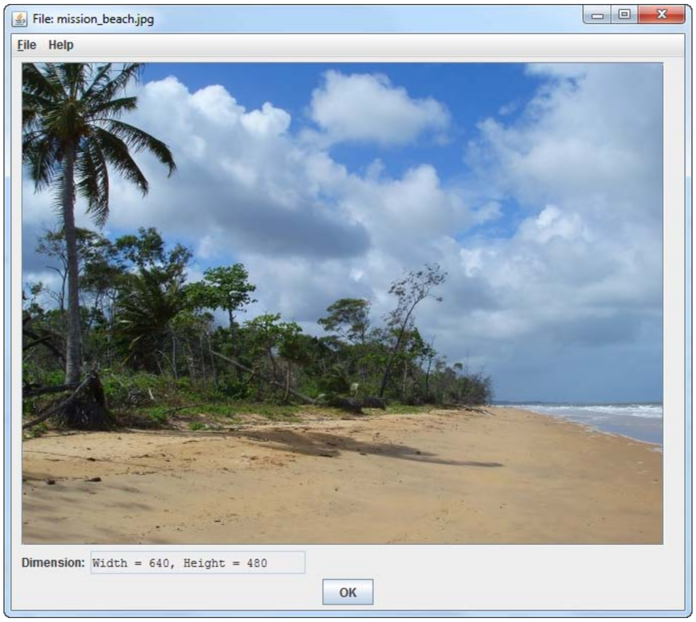
Edges and Echoes

Due: Nov. 18, 2016 @ 4:00 pm (late date Nov. 21 @ 4:00 pm)

The emphasis for this assignment is processing a pictures and sounds as collections. In preparation for this assignment, create a folder called `Assign_5` for submission of the assignment and two subfolders: `Assign_5_A` and `Assign_5_B` for the two DrJava projects for the assignment.

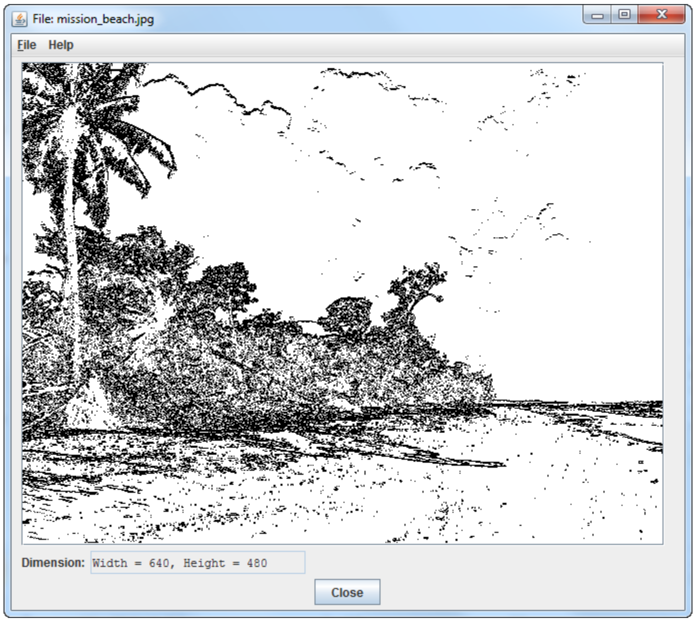
Part A – Edge Detection

Write a Java program to do simple edge detection on a picture. The resulting image will have black pixels wherever there is an edge in the original picture and white pixels elsewhere. For example, the picture:



Dimension: Width = 640, Height = 480

after edge detection looks like:



Dimension: Width = 640, Height = 480

revised: 11/11/2016

To do edge detection: on each row we compare the intensity of the pixel with the pixel immediately below it (i.e. on the next row). If the absolute difference in the intensities is smaller than a value `TOLERANCE` (a constant with value `10.0`), we set the pixel to `white`, otherwise we set it to `black`. Since the last row has no row below it, we treat it as if the row below it is the same (i.e. all the intensities are the same).

Use a helper method:

```
     private double intensity ( Color c ) {…
```

that returns the intensity of the `Color c`, defined as the average of the R, G and B components.

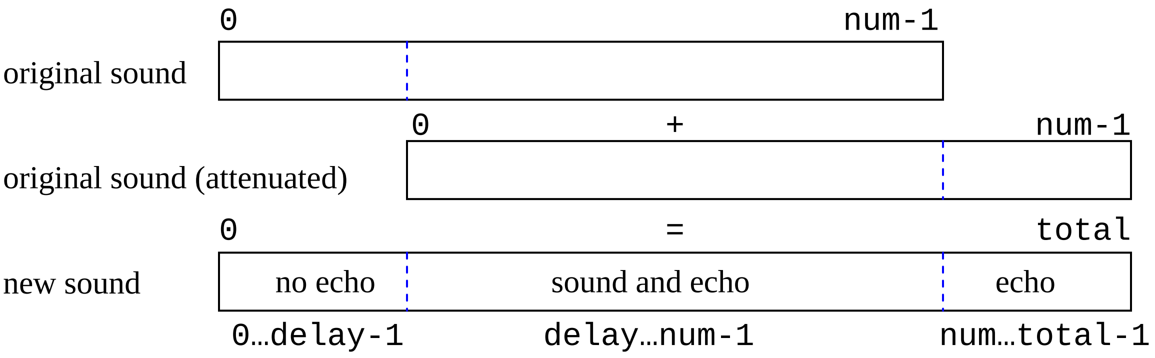
Part B – Echo

An echo occurs when sound waves produced at one place reflect off a hard surface and return back to the original source. The time it takes for the sound to reach the surface and then return is the delay in the echo. Since sound attenuates (diminishes) as it travels, the reflected sound (echo) is quieter than the original.

Write a method:

```
     private Sound echo ( Sound aSound, double delay,
                          double factor ) {…
```

which produces a sound with an echo based on the original sound `aSound` with a delay of `delay` seconds and an attenuation of `factor`. The delay can be converted to a number of samples by multiplying by the sampling rate of the sound. The new sound will be longer than the original sound by the number of samples that make up the delay. The new sound will have three parts. In the first part, lasting as long as the delay, there is no echo and the samples in the new sound will just be a copy of the samples from the original sound. Once the delay has elapsed, the second part consists of the blending of the original sound and the echo. The samples of the new sound will be the sum of the amplitudes for the remaining (after the delay) samples of the original sound plus the amplitudes (attenuated by the attenuation factor) of the samples of the original sound starting from the beginning (the echo). Finally, the last part of the new sound will be just the remaining echo. The samples will be the remaining samples of the original sound (i.e. where we left off with the echo in the second part) at the attenuated amplitude. This correspondence is shown in the diagram below where `delay` is the number of samples in the delay, `num` is the number of samples in the original sound and `total` is the number of samples in the new (echo) sound (equal to `delay +num`).



Write a program that loads a sound, allows the user to play it and then produces a sound with an echo having a delay of `0.5` seconds and an attenuation of `0.25`. The new sound should then be presented for the user to play and, when done, the new sound should be saved.

Submission:

Details regarding preparation and submission of assignments in COSC 1P02 are found on the COSC 1P02 Sakai Site as `Assignment Guidelines` under `Course Documents`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below.

To prepare and submit the assignment electronically, follow the procedure below:

1. Ensure your submission folder (`Assign_5`) has two subfolders (`Assign_5_A` and `Assign_5_B`) that contain the two DrJava projects for the assignment.
2. Using DrJava, print (to `CutePDF Writer`) the `.java` file of each of the parts for your assignment using the name *`ClassName`*`.pdf` where *`ClassName`* is the class name (i.e. same name as the `.java` file) and save the `.pdf` files at the **top level** of the submission folder (i.e. directly within `Assign_5`).
3. Run each program. For Part A use the file `mission_beach.jpg`. When the program presents the save dialog, save the resulting picture file at the **top level** of the submission folder (i.e. directly within `Assign_5`) using an appropriate name (e.g. `Part_A.jpg`). For Part B use the file `thisisatest.jpg`. When the program presents the save dialog, save the sound file at the **top level** of the submission folder (i.e. directly within `Assign_5`) using an appropriate name (e.g. `Part_B.wav`).
4. Create a `.zip` file of your submission by right-clicking on the top level folder (i.e. `Assign_5`) and selecting `Send to/Compressed (zipped) folder`. A zipped version of the folder will be created. Use the default name (`Assign_5.zip`).
5. Log on to Sakai and select the COSC 1P02 site.
6. On the `Assignments` page select `Assignment 5`. Attach your `.zip` file (e.g. `Assign_5.zip`) to the assignment submission (use the `Add/Remove Attachments` button and select `Browse`). Navigate to where you stored your assignment and select the `.zip` file (`Assign_5.zip`). The file will be added to your submission. Be sure to check the `Honor Pledge` checkbox. Press `Submit` to submit the assignment. You should receive a confirmation email.

DrJava

The `.zip` folder you submit should contain the project folders for the two parts, including all files relevant to the project—the `.java` and `.class` files for the assignment—and the `.pdf` files for program listings and output at the top level.

Other Platforms

If you are using an IDE other than DrJava to prepare your assignment, you must include the `.java` source files and the `.pdf` files described above for each part as well as an executable file (likely `.class` or`.jar`) that will execute on the lab machines.